



La compra fácil

Memoria de Proyecto Final de Grado

Grado en Multimedia

Desarrollo en Aplicaciones Interactivas

Autor: Jose Luis Samitier Aliaga

Consultor: Kenneth Capseta Nieto

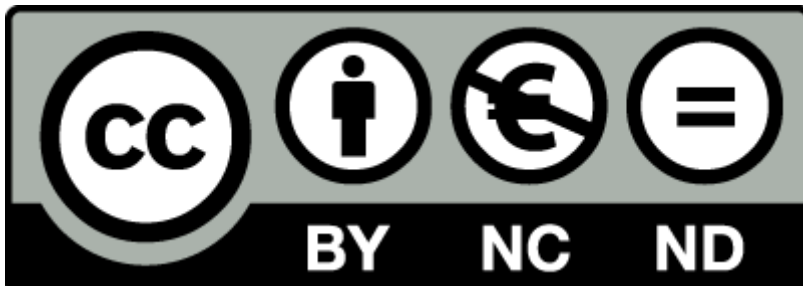
Profesor: Carlos Casado Martínez

Fecha de entrega: 18 / 6 / 2017

Créditos / Copyright.

Licencia:

Esta obra está sujeta a una licencia de:



Reconocimiento–No Comercial–Sin obraDerivada3.0 España de Creative Commons.

(<http://creativecommons.org/licenses/by-nc-nd/3.0/es/>)

Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia.

Bajo las condiciones siguientes:

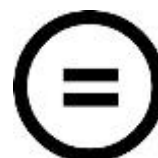


Reconocimiento — Debe reconocer adecuadamente la autoría, proporcionar un enlace a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciador o lo recibe por el uso que hace.



No Comercial — No puede utilizar el material para una finalidad comercial.

Sin Obra Derivada — Si remezcla, transforma o crea a partir del material, no puede difundir el material modificado.



No hay restricciones adicionales — No puede aplicar términos legales o medidas tecnológicas que legalmente restrinjan realizar aquello que la licencia permite.

Abstract.

Lo incluido en este documento se corresponde al Proyecto de Fin de Grado en Multimedia en 'Desarrollo de Aplicaciones Interactivas' de la Universitat Oberta de Catalunya.

Este proyecto consiste en una aplicación para gestionar nuestras compras de alimentación en grandes superficies. Esta información la podremos manejar a través de una página web adaptable (responsive), que nos permitirá acceder desde nuestro ordenador personal, una Tablet o un Smartphone.

La idea de este proyecto ha surgido al revisar las aplicaciones similares existentes en el mercado. En todas ellas debemos acceder a partir de un artículo. Esta aplicación nos ayudará a crear una lista de compra. Y, a partir de ella, nos sugerirá donde comprar: podrá ser en un único establecimiento o nos sugerirá que dividamos dicha compra en un máximo de tres.

El objetivo de esta aplicación consiste en facilitar la compra suministrando información de los lugares más baratos, más cercanos o, incluso, se da la posibilidad de realizar la compra on-line, por medio de un comparador de precios.

Esta aplicación contendrá una base de datos que contendrá la información sobre establecimientos, productos, usuarios y sus preferencias, así como de configuración. A la que se accederá por medio de un formulario para obtener una tabla con el resultado de los productos seleccionados, sus precios y los establecimientos.

Palabras clave: Comparador, Supermercados, Lista de Compra, Trabajo de Final de Grado, Precios

Notaciones y Convenciones.

Tipografía

Para la elección de la tipografía se han consultados diferentes guías de estilo, algunas de las cuales se especifican en la bibliografía. En la decisión sobre la tipografía a elegir se han tenido en cuenta los estándares propuestos, que pueden producir una cierta contradicción:

- Para los textos escritos (extensos) se recomienda una tipografía de tipo 'serif', como puede ser 'Baskerville', 'Bodoni', 'Caslon', 'Century Old Style', 'Fournier', 'Garamond', 'Sabon' y sobre todo 'Times New Roman'.
- Y para la web se recomiendan del tipo 'sans serif' como, por ejemplo: 'Arial', 'Avenir', 'Century Gothic', 'Frutinger', 'Gill Sans', 'Helvética', 'Lucinda Sans', 'Myriad', 'Tahoma', 'Trebuchet', 'Univers' y 'Verdana'.

Sin embargo, nuestro TFG es un producto mixto al ser pensado principalmente en ser distribuido en formato PDF, para su consulta en ordenador y se ha utilizado una tipografía Arial.

- Tipo de letra de todo el cuerpo del proyecto: Arial.
- Tipografía para títulos y subtítulos: Arial.
- Espaciado: 1,5 líneas.
- El subrayado, cuando se requiera será sencillo.

NOTA: la tipografía corporativa de la UOC (UOC Sans) no está disponible.

Tipo de página y márgenes

El proyecto estará escrito en formato A4 según norma UNE 1011, por una sola cara, con los siguientes márgenes:

- izquierdo, 35 mm.
- derecho, 15 mm.
- Superior, 25 mm. e
- inferior, 25 mm.

Título Arial 20, negrita.

- Entre el nombre del título y el comienzo del primer párrafo habrá un espaciado adicional

Subtítulo Arial 15.

- Entre el fin del último párrafo de un apartado y el comienzo de otro apartado o de un sub-apartado habrá un espacio adicional de 9 puntos.

Subsección Arial 12, subrayada sencilla.

- El título del sub-apartado o división de tercer nivel se escribirá subrayado, en minúsculas.
- Entre el fin de un sub-apartado y el comienzo de un apartado o de otro sub-apartado habrá un espacio adicional de 9 puntos

Cuerpo Arial 10.

Título de los anexos Arial 20, negrita.

Subtítulo Arial 15.

Índice

Créditos / Copyright.....	2
Abstract.....	3
Notaciones y Convenciones.....	4
Tipografía.....	4
Tipo de página y márgenes.....	5
Índice.....	6
1. Introducción.....	8
2. Descripción.....	9
3. Objetivos.....	11
3.1 Principales.....	11
3.2 Secundarios.....	11
3.3 Propuestas de ampliación.....	12
4. Escenario.....	13
5. Contenidos.....	16
6. Metodología.....	19
7. Arquitectura de la aplicación.....	20
8. Plataforma de desarrollo.....	21
9. Planificación.....	24
10. Proceso de trabajo.....	26
11. APIs utilizadas.....	31
12. Diagramas.....	32
12.1 Diagrama Entidad / Relación.....	32
12.2 Flow Chart – Creación de una Lista de Compra.....	33
12.3 Casos de uso.....	34
13. Prototipos.....	41
13.1 Lo-Fi - Hi-Fi.....	41
14. Perfiles de usuario.....	45
15. Usabilidad.....	47
16. Seguridad.....	50
17. Test.....	51
18. Versiones de la aplicación/servicio.....	52
19. Requisitos técnicos.....	53
20. Instrucciones de uso.....	54
21. Bugs.....	57
22. Proyección a futuro.....	58
23. Presupuesto.....	59
24. Análisis de mercado.....	60
25. Marketing y Ventas.....	61
26. Conclusión.....	62

Anexo 1. Entregables del proyecto.....	63
Anexo 2. Diagrama de navegación	64
Anexo 3. Entidades y Relaciones.....	65
Anexo 4. Código fuente (extractos)	68
Anexo 5. Capturas de pantalla	78
Anexo 6. Resumen ejecutivo.....	84
Anexo 7. Bibliografía	85

1. Introducción.

Con este proyecto se pretende realizar una propuesta que aporte más funcionalidades y una mejor experiencia de usuario a los actuales buscadores de precios para artículos de supermercados. Los comparadores de precios existentes sólo permiten la comparación productos a producto, lo cual los hace muy limitados en cuanto a usabilidad. Ya que se debe buscar el mejor precio artículo a artículo y después, realizar una lista manual y decidir en qué establecimiento se efectuará la compra total. Son útiles para conocer el mejor precio de un artículo, pero no para efectuar una lista de compra.

Se pretende ofrecer una aplicación cerrada que ofrezca las funcionalidades imprescindibles para realización de su función. Será una aplicación modular que se pueda implementar por partes. Y que permita ser capaces de tener la seguridad que realizamos la mejor compra posible dentro de los condicionantes existentes: precio, localización de los establecimientos, recorrido óptimo (no se pretende comprar un artículo en cada supermercado).

Para que exista una competencia real, debemos tener una información veraz. O, al menos, acercarnos tanto como podamos. Si se dispone del mismo artículo más barato en un establecimiento que en otro, lo lógico sería comprar en el más barato. El problema aparece cuando no todos los productos están más baratos en el mismo sitio, sino que se encuentran en diferentes supermercados. Decidir a qué establecimiento se debe ir, no es fácil. Sí lo es, si tenemos en cuenta sólo un solo centro, pero, y si queremos ir a dos, o a tres, ¿cómo decidimos?



Figura 1: <https://articulos.softonic.com/apps-comparar-precios>

2. Descripción.

Este proyecto intenta la realización de una aplicación que nos ayude a la realización de nuestras compras en los grandes supermercados, con la garantía de que es el cliente el que decide comprar un producto determinado en un centro concreto en base a indicadores: precio, cercanía, garantía...

A su vez, se pretende que el diseño o funcionalidad no se limite a un aspecto tecnológico. La idea, que no exactamente la aplicación, se podrá utilizar desde un terminal de ordenador, desde una Tablet o desde un Smartphone, e, incluso, podrá generar una información en papel. Todo ello, para no limitar, dentro de lo posible, a ningún usuario por temas de familiaridad técnica o habilidad. Aunque, por otra parte, podrá ofrecer a aquellos usuarios avanzados por conocimiento o por disponibilidad de equipos más versátiles (tabletas o Smartphone de altas prestaciones), una experiencia de usuario suficientemente atractiva.

La diferencia principal con los buscadores actuales, reside en que en éstos se busca producto a producto y se accede a diferentes bases de datos. En esta propuesta, se parte ya de una lista hecha que accederá por lotes. Los comparadores actuales obligan a realizar las listas a partir de una relación de productos, con unos accesos producto a producto, para después obtener los precios más reducidos. Pero, a partir de ahí, es poco práctico tomar decisiones si la lista tiene muchos artículos. El prototipo que se presenta, con las ayudas para realizar la lista, que, a su vez, sirve de recordatorio, obtiene los mejores precios y propone soluciones: a que centro o centros ir.

La aplicación pretende servir de ayuda a cualquier persona que quiera efectuar sus compras en grandes supermercados de alimentación, de una forma más intuitiva, eficaz y económica. El punto de partida será la realización de la lista de compra, lo que hace diferente esta aplicación de las existentes en la web. La lista, en la versión final del proyecto, podrá hacerse de diferentes formas:

- Introduciendo por texto el artículo.
- Obteniéndolo por captura de código de barras.
- Por captura de código QR.
- Por selección guiada: tipo de artículo y selección de una Base de Datos.
- Por notas de voz.
- Por recuperación de una lista almacenada en *favoritos*.

Una vez confirmada la lista de compra, se accederá a la base de datos de la aplicación. Esta BDD centralizada se actualizará de dos formas:

- Incorporando los artículos directamente.
- O de los centros comerciales – opción, por supuesto, sujeta a la disponibilidad de obtener directamente la información de los propios centros. seleccionados si se desea, para obtener los precios de compra.

Desde la BDD se obtendrá toda la información necesaria: precios, localización de centros, artículos...

Con la información obtenida se podrán elegir las diferentes propuestas de compra con sus precios asociados:

- Por un único centro comercial: propone el centro más cercano de la marca comercial elegida, conociendo la localización del usuario. El usuario podrá elegir: más barato, más cercado, de más confianza...
- Un recorrido por 'n' centros a partir de la localización actual. El usuario, de nuevo, podrá elegir.
- Accediendo a la web del establecimiento y efectuando compra on-line, para uno o varios centros.
- Para artículos específicos, por precio o por ser muy concretos, se podría acceder al web on-line de Amazon, Rakuten, eBay. (NO SE IMPLEMENTA EN ESTE TRABAJO FINAL).

Esta web nace de cero y la propuesta presentada contempla una parte de sus funcionalidades finales debido a su complejidad. Se implementa utilizando HTML5, CSS3, PHP y una base de datos MySQL. Asimismo, se utilizan funcionalidades de Javascript, Ajax y algo de jQuery.

3. Objetivos.

3.1 Principales.

Objetivos clave del TF.

- Realizar una aplicación beta para ordenador con bases de datos de pruebas. No se plantea la conexión con los establecimientos.
- Diseñar prototipos que puedan utilizarse también en una Tablet (13 o 15 pulgadas).
- Efectuar el diseño de funcionamiento para cualquier tipo de interface en base a una aplicación responsive.
- Aplicar los conocimientos aprendidos durante el Grado.
- Funcionalidades implementadas:
 - Diseño gráfico de la aplicación.
 - Alimentar la base de datos que permita la utilización en pruebas.
 - Establecimientos, productos, usuarios, favoritos
 - Creación de algoritmos dinámicos que permitan una búsqueda de productos
 - Por categorías, subcategorías, y aplicando filtros por productos y marcas.
 - Gestión de login y registro de usuarios
 - Crear lista de compra.
 - Se podrá modificar la lista actual, eliminando o añadiendo productos, así como guardarla en favoritos
 - Imprimir propuesta de compra.

3.2 Secundarios.

Objetivos adicionales que enriquecen el TFG y que pueden sufrir variaciones.

- Crear listas de compra favoritas.
- Crear algoritmo de decisión de compra
- Funcionalidades a implementar:
 - Mostrar localización establecimientos almacenados en base de datos de prueba.
 - Acceso a las tiendas on-line de los establecimientos
 - Control de la seguridad de la BDD
 - Utilización de AJAX para mejora de usuario.

3.3 Propuestas de ampliación.

Objetivos adicionales que se pueden implantar en un futuro para cerrar la aplicación.

- Utilización de la geolocalización
- Búsqueda de productos:
 - Por código de barras, código QR, notas de voz
- Creación de una base de datos histórica de artículos consultable por:
 - Fecha, establecimiento, artículo.
- Creación de un histórico de gastos realizados.
- Según disponibilidad de los establecimientos: alimentación automática de la BDD directamente de la información de los establecimientos.

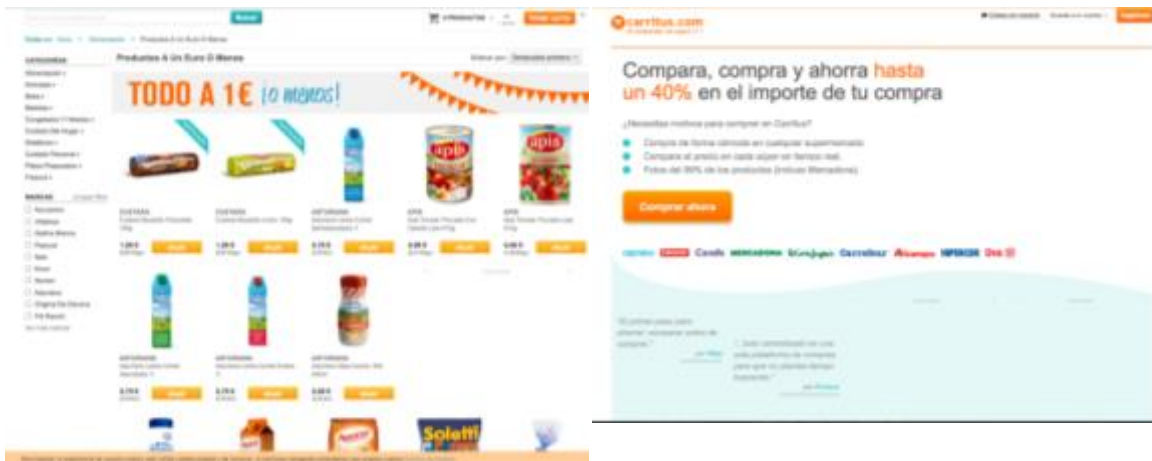
4. Escenario.

En la actualidad, se considera que es muy difícil encontrar una idea original que solucione cualquier problema cotidiano. En este caso, se aborda el tema de los comparadores que existen actualmente. De los más conocidos, los de viajes y hoteles. Esta aplicación se centra en un comparador de precios de artículos de supermercado. No tan desarrollados como los anteriores, quizás por la ingente variedad de artículos diferentes, del acceso a sus bases de datos de precios, de lo que se llamarán ofertas (bien por cantidad, por temporalidad, por bonus de las marcas suministradoras, o por productos llamados 'de marca blanca', entre otras peculiaridades).

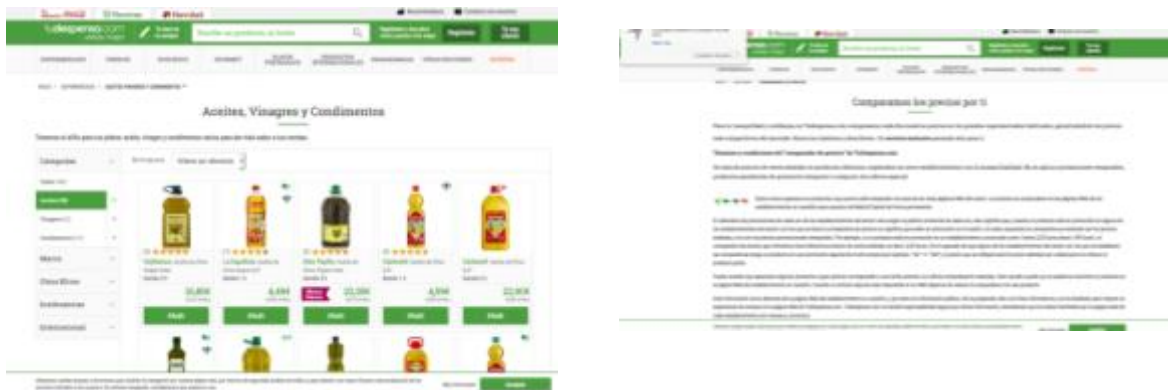
Se tiene que tener en cuenta no penalizar el tráfico de datos de los usuarios que utilicen Smartphone al acceder a la base de datos de la aplicación. Como en los temas de localización. Se está diseñando una aplicación que corra principalmente en un ordenador doméstico que obtenga los resultados y que éstos se puedan transferir o modificar a los Smartphone o tabletas.

A continuación, se proponen algunos ejemplos de aplicaciones que existen en la actualidad:

- 1) Carritus (<http://www.carritus.com/>)



- 2) Tu despensa (<https://www.tudespensa.com/ventajas/comparamos-los-precios/>)



3) Organización de consumidores:

<http://www.ocu.org/consumo-familia/supermercados/calculadora/precios-supermercados>



Esta aplicación sólo muestra una comparativa global entre establecimientos a partir de una lista única.

No es funcional para poder comparar una lista de compra creada por un usuario.

4) Komparing: <https://www.komparing.com/es/supermercados/>



Esta aplicación solo permite saber el precio de un producto en concreto y lo relaciona con el establecimiento más barato.

Al final, sólo tenemos una lista de productos relacionados con los 'n' establecimientos establecidos. Lo que tampoco la hace funcional para un usuario, ya que no va a ir a cada uno de los establecimientos a comprar pocos productos en cada uno de ellos.

5) Soy súper, tu supermercado on-line <https://soysuper.com/>



En esta aplicación se puede comprobar un funcionamiento análogo: se elige un establecimiento favorito y después elegimos los productos, uno a uno

Se puede ver en todas las webs indicadas, que el funcionamiento es muy similar:

- se accede por producto,
- se compara el precio,
- se indica, a veces, el establecimiento, y
- tú te haces la lista.

En la propuesta que se presenta, el funcionamiento es diferente desde el inicio.

No se pretende ir comparando un precio producto a producto, sino que el punto de inicio es la creación de una lista de compra. Que, en definitiva, es el punto inicial del proceso.

Un usuario potencial, lo primero que hace es crear su lista de la compra. Bien toda seguida, en un proceso periódico (compra semanal), con una lista favorita o completamente nueva. O se va creando cada vez que se produce un suceso (una falta de producto o un recordatorio).

Este punto es importante, sobre todo cuando se implanten todas las funcionalidades de la aplicación. Especialmente la versión móvil y la captura por código de barras, código QR o por voz. Ya que la lista estará sincronizada en la web para todos los dispositivos de ese usuario (que deberá estar registrado).

Así, una vez la lista creada, podrá consultar el precio más barato para cada producto y para cada establecimiento. Lo que le dará un poder de decisión de elegir el supermercado más barato de la lista total, o podrá dividir esa lista en partes. E incluso plantear una compra física en su sucursal más cercana y el resto on-line.

5. Contenidos.

Para realizar este proyecto, primeramente, se debe 'tener la idea'. En este caso, aparece por generación espontánea. Se pensaba de inicio, en realizar una página web de un establecimiento. A partir de este punto, por comparación con unos buscadores, se pensó en la posibilidad de realizar un comparador para productos de supermercados y grandes superficies. A continuación, se hizo una búsqueda de productos parecidos (siempre los hay) para comprobar lo que ofrecían. Se tenían claros varios conceptos:

- Que la sistemática que se ofrece pasa por averiguar producto a producto su precio y que tener la constancia de cual era más barato, era 'poco amigable'.
- Que se quería partir de una Lista de compra, a partir de la cual, se propondrían las acciones a realizar.
- Que se quería tener un producto cerrado, con todas las opciones:
 - o Compra física / compra on-line.
 - o Todo tipo de usuario: poco hábil y experto.
 - o Todo tipo de terminal: ordenador, tableta y Smartphone.
- Minimizando los recursos necesarios: mínimo consumo de datos. Lo que significaría que parte del 'trabajo duro' se podría hacer en el ordenador, para pasar después a los otros terminales.
- Se tenían muchas dudas sobre la obtención de la información de los establecimientos, así como su complejidad, en base a ofertas, marcas blancas, etc.
- Y que, dada la complejidad y cantidad de los datos que se manejan, no nos planteamos un proyecto 'mobile first'. Creamos una página web que después ajustaremos a las dimensiones de una tableta o Smartphone. Sin embargo, se piensa, en un inicio, utilizar Bootstrap para el diseño, lo que nos ayudará después a dicho ajuste. Es decir, creamos una página web, pero teniendo siempre en mente, que crearemos un diseño para Smartphone al final.

Por medio del benchmarking de la competencia y de las ideas que se querían realizar, se han definido una serie de requerimientos que se quieren abordar. Y se pretende realizar en base a un Diseño Centrado en el Usuario (DCU). Se utilizará para esta interfaz, un estilo contextual, con ventanas de dialogo, menús (básicamente, desplegados, jerárquicos o en cascada) y campos de formulario (ventanas de selección: cajas de texto, checkbox).

- 1) Se pretende una página de inicio que contendrá:
 - a. Un espacio lateral izquierdo que utilizaremos para efectuar una primera selección de artículos.
 - b. Otro espacio lateral derecho que se mantendrá oculto y se utilizará en una fase posterior, cuando la aplicación esté completamente terminada, para tareas de mantenimiento. Accesible por contraseña de administrador.
 - c. Una cabecera propia donde incluiremos
 - i. El enlace para registrarse
 - ii. El enlace para identificarse
 - iii. El identificar de usuario que usa la aplicación.
 - iv. Un cambio de idioma (Futuro)
 - d. Un footer.
 - e. La información se distribuirá en una zona central, planteada para utilizar con Bootstrap:
 - i. Un cuadro de texto para buscar por descripción
 - ii. Un espacio para filtros.
 - iii. Un espacio para filtrar por marca.
 - iv. Un espacio para presentar los productos encontrados.
- 2) Y una interactividad:
 - a. Se creará un registro en la lista por:
 - i. Una selección de filtros: categorías de producto, subcategoría, filtros y marcas.
 - ii. Teclado: se irán proponiendo nombres a partir de lo ya escrito o por selección de las propuestas obtenidas por los filtros (Futuro).
 - iii. Código de barras (Futuro).
 - iv. Código QR (Futuro).
 - b. Función validar lista:
 - i. Acceso a las BDD de los establecimientos seleccionados (por personalización). Se elegirán 'n' establecimientos de los existentes.
 - ii. Modificará las columnas 'categorías', marcas' y 'productos' por tantas columnas como establecimientos tengamos seleccionados (hasta un máximo a determinar), en los que se informará de los precios de los artículos.
 - iii. Cuando no exista ese producto, por defecto lo dejará en blanco, con la posibilidad de volver a la selección para elegir otro producto parecido de ese establecimiento concreto y se creará línea nueva.
 - iv. Visualizará los totales por establecimiento.
 - c. Función guardar lista en favoritos.
 - d. Función imprimir lista en PDF.

- e. Función decisión lista donde decidiremos cómo queremos comprar con esa lista:
 - i. Compra física marcando 1 o 2 establecimientos.
 - ii. Podrá proponer los centros más cercanos (introduciendo Código Postal).
 - iii. Compra on-line a los centros seleccionados.
 - f. Función guardar decisión lista: la hará accesible desde tableta o Smartphone.
 - g. Función activar lectura código de barras (Futuro)
 - h. Función activar lectura código QR (Futuro).
 - i. Función registrarse: permitirá el acceso a listas favoritas e información de envíos pendientes, si los hubiera.
- 3) La BDD será interna y se creará a través de formulario. Se pre supone la dificultad de obtener la información directamente de los establecimientos.
- 4) Se tendrá acceso a una segunda página de información histórica del usuario:
- a. Compras realizadas en el tiempo, por establecimientos, por tipo.
- 5) Conceptos importantes:
- a. En esta primera fase del proyecto, se desconoce el tamaño y las posibilidades de acceso a las BDD de los centros comerciales. Se crearán unas BDD de trabajo para las pruebas.
 - b. El diseño y la maquetación se plantea en una etapa posterior, una vez se hayan probado las funcionalidades.
 - c. No se plantea, en esta primera versión, disponer de páginas en diferentes idiomas.

Como hemos comentado, se está planteando este proyecto como un proyecto 'redondo' con muchas actualizaciones, lo que lo hace dimensionalmente muy extenso para un Trabajo Final de Grado. Por ello, no se realizarán todas las funcionalidades, aunque se tendrán en cuenta en el diseño.

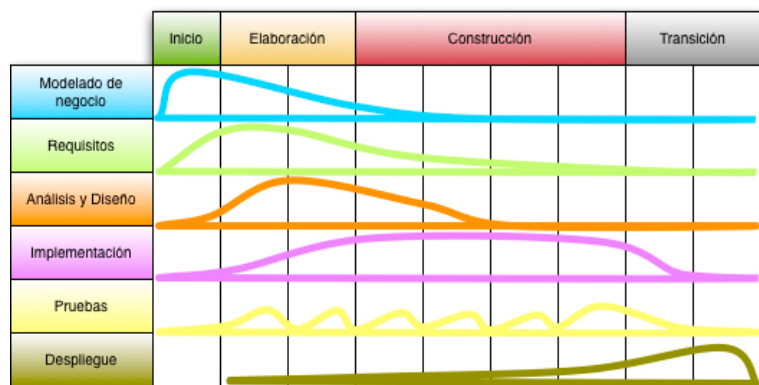
6. Metodología

Una de las tareas a realizar es la de elegir el método de desarrollo más adecuado. En este proyecto se hace frente a:

- Una solución conocida: conocemos las partes, pero no dominamos cómo funcionarán en conjunto.
- Un objetivo claro.

Se decide que el método de desarrollo sea el 'Ciclo de vida incremental'. Y se plantea un tipo de 'proceso unificado', que se basará en diferentes implementaciones que se irán realizando progresivamente. Cada una de ellas con las siguientes etapas:

- Definición de requisitos.
- Análisis y diseño.
- Implementación.
- Pruebas.
- Mantenimiento.



Y se procurará dejar operativas cada una de las iteraciones.

El proyecto pasará por las siguientes fases:

- Inicio.
- Elaboración.
- Construcción.
- Transición.

Y se plantearán las siguientes actividades:

- Modelización del negocio: se establecen los casos de uso.
- Requisitos: que debe hacer y que no debe hacer el sistema.
- Implementación: escribir y verificar el código.
- Pruebas.
- Deployment: gestión de las versiones.

Dada la magnitud del proyecto, en su dimensionamiento para este TFG, los roles que se plantean tener (Stakeholder, Jefe de Proyecto, Analista, Arquitecto, Desarrollador y Experto en pruebas) se centran todos en el realizador de dicho TFG.

Sin embargo, se podrían implementar si se realiza el proyecto en su totalidad.

7. Arquitectura de la aplicación

Es un proyecto de una página web que se podrá consultar en el navegador del usuario, con criterios responsivos y que se utilizará un servidor web ubicado en un hosting.

El hosting empleado corresponde al proveedor 'Webempresa'. Se realizó una comparativa entre diferentes proveedores y se consideró que era el más adecuado, por prestaciones y por servicio.

En un principio, se decidió utilizar un framework de desarrollo llamado Laravel. Sin embargo, a medio proyecto se tomó la decisión de efectuar el desarrollo creando las páginas directamente, lo cual afectó al timing y objetivos del mismo. Esta decisión fue motivada por el tiempo necesario para conocer el funcionamiento del framework en sí, y no por sus posibilidades. Se considera a Laravel un producto de muchas posibilidades y muy útil para construir este tipo de proyectos, pero que, evidentemente, necesita un aprendizaje y unas herramientas de desarrollo adecuadas.

Ya que el sitio web debe mantener un esquema estable de diseño y también, para facilitar el desarrollo y mantenimiento de las páginas, éstas se han construido por partes y se han unido con 'includes':

- Partes comunes a todas las páginas:
 - o Un header – que incluye los controles de las barras laterales, los accesos a login y registros y la identificación del usuario.
 - o Un footer
 - o Una barra lateral Derecha de selección inicial de productos.
 - o Una barra lateral izquierda de mantenimiento y de gestión de usuario
 - o Una barra de navegación de botones.
- Una parte central de contenido que se dividirá en dos partes que se utilizarán en diferentes páginas y que mantendrán el estilo de ellas.



8. Plataforma de desarrollo

Para desarrollar el proyecto, se ha utilizado:

Software:

- a) Para desarrollo de la memoria - Microsoft Office 365 (2016)
 - a. Word para Mac (versión 15.32 Office 2016 Mac) – procesador de textos.
- b) Wireframes / Prototipado
 - a. Axure RP8 – (Team Edition 8.0.0.3333) Es una herramienta de prototipado profesional que permite crear wireframes para hacer el pre-diseño de una página web. Disponible tanto para plataforma Windows como Mac. Se pueden dibujar tanto wireframes estáticos como interactivos que simulan una experiencia de navegación del usuario real. Tiene una función de colaboración para compartir las maquetas Web con otras personas y poder recibir feedback en tiempo real.
 - b. PowerPoint (versión 15.32 Office 2016)– bajo nivel
 - c. Adobe Illustrator (Master Collection CS6)
- c) Planificación del proyecto
 - a. Microsoft Project Profesional 2016 64 bits
- d) Pruebas en navegadores
 - a. (el trabajo de base se realiza en un iMac de sobremesa, con OS Sierra y en portátil MSI con Windows 10)
 - b. Mozilla – Firefox 50.1.0
 - c. Safari – Versión 10.0.2
 - d. Google Chrome – Versión 56.0.2924.79
- e) Video de presentación - Adobe Creative Suite CS6
 - a. Premiere (Master Collection CS6) – montaje del video de presentación
 - b. After Effects (Master Collection CS6) – montaje y revisión del video de presentación.
- f) Editores de código
 - a. Dreamweaver (Master Collection CS6)
 - b. Sublime Text – versión 3
 - c. Notepad ++ (versión 7.3.3) X64 bits
- g) Máquinas virtuales
 - a. Para el proceso de desarrollo, se utiliza una máquina virtual en un ordenador portátil MSI con Windows 10. Para ello se instala en la máquina Windows:
 - i. Git para Windows v2.12.0– para usar como terminal en lugar de la de Windows

- ii. Vagrant (Versión 1.9.3) – para instalar la ‘box’ de Lavarel-Homestead que se usará como entorno de desarrollo
 - iii. Virtual Box de Oracle (Versión 5.1)
- b. Laravel Homestead (como se ha dicho), que trabaja en un entorno virtual con Vagrant / Virtual Box. Este entorno virtual lleva instalado:
- Ubuntu 16.04
 - Git
 - PHP 7.1
 - Nginx
 - MySQL – Base de Datos elegida
 - MariaDB
 - Sqlite3
 - Postgres
 - Composer
 - Node (With Yarn, Bower, Grunt, and Gulp)
 - Redis
 - Memcached
 - Beanstalkd
 - Mailhog
 - Ngrok
- h) Para la segunda parte del proyecto (ya sin Laravel), se crea un Servidor Apache en un XAMPP
- i) Gestor de bases de datos –
- a. MySQL Workbench (Versión 6.3) - herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL

Hardware:

Sobremesa

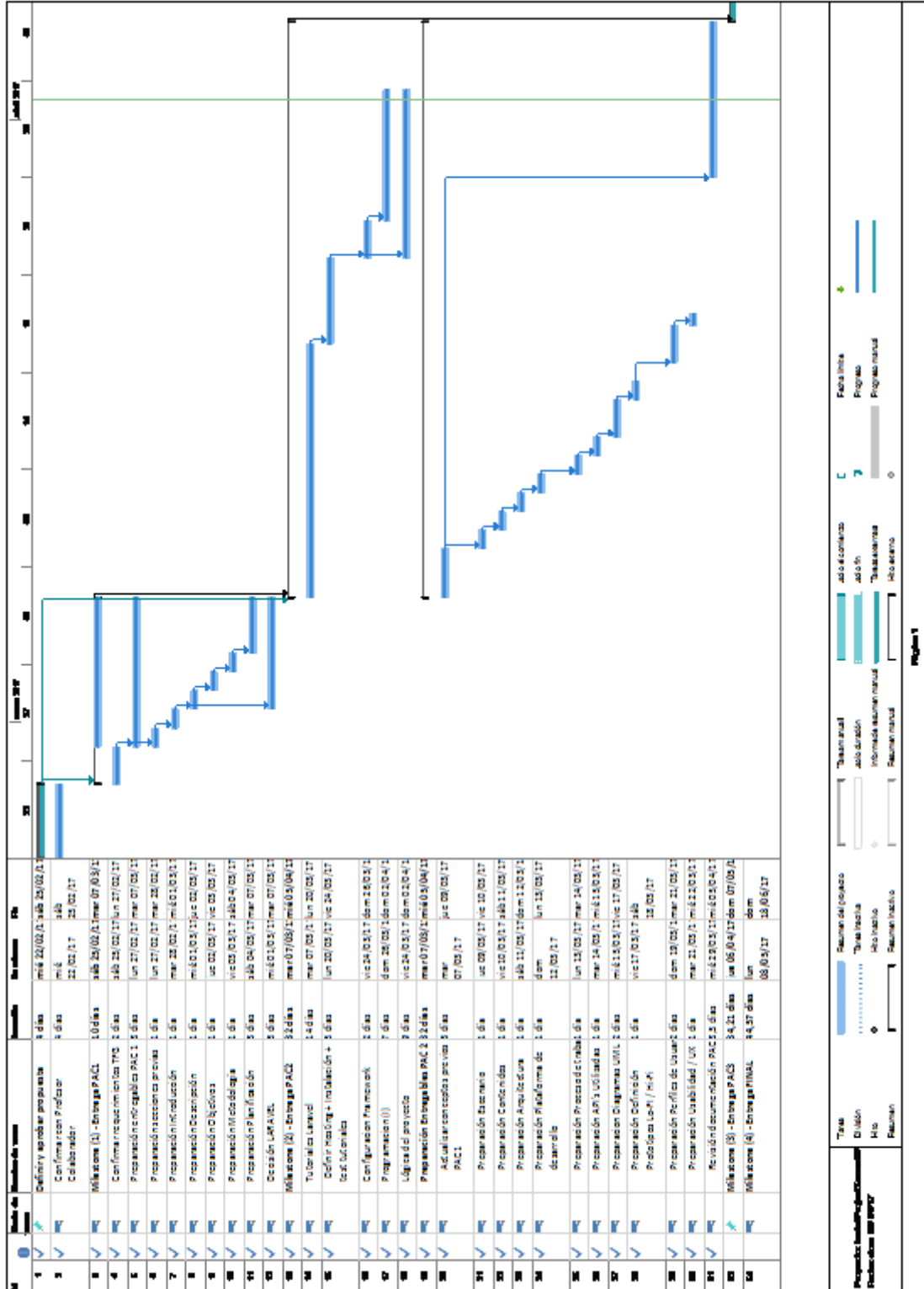
- iMac 21,5 finales 2012
- Sistema operativo: OSX Sierra 10.12.2
- Procesador: Intel i5 @ 2.70 GHz
- RAM: 16 GB, 1600Hz DDR3
- Disco Duro: 500 GB SSD
- Pantalla: Full HD (1920 x 1080)
- Tarjeta gráfica: NVIDIA GeForce GT 640M 512 MB.

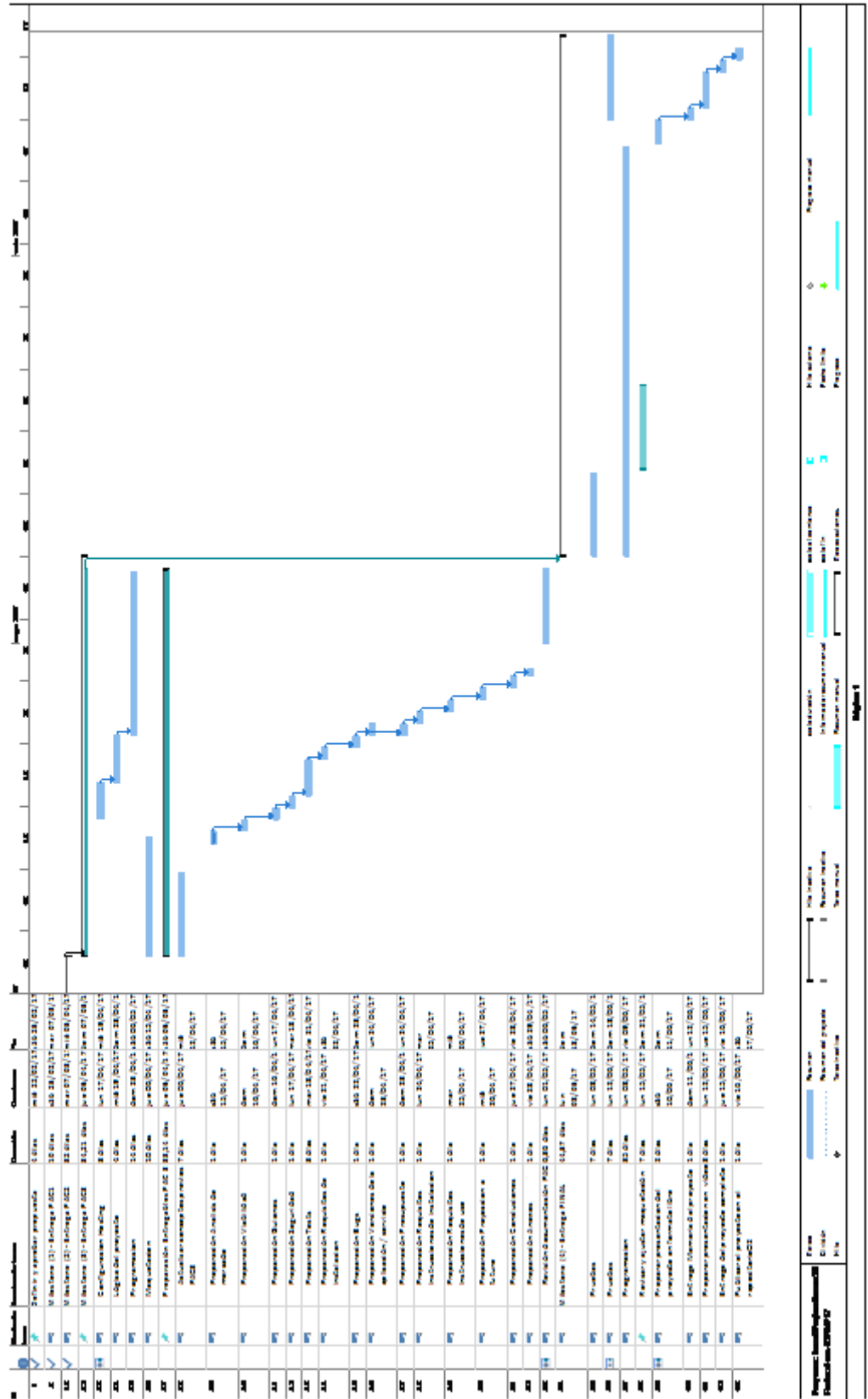
Portátil:

- MSI – GE62 6QD
- Sistema operativo: Windows 10
- Procesador: Intel i7-6700 HQ @ 2.60 GHz
- RAM: 16 GB, 1600Hz DDR3
- Disco duro: 256 GB SSD + 1 Tb 7200 rpm
- Pantalla: Full HD (1920 x 1080)
- Tarjeta gráfica: NVIDIA GeForce GTX 960 M con 2Gb.

9. Planificación

Anexo primer borrador del diagrama de Gantt para el proyecto





10. Proceso de trabajo

El proceso de trabajo se ha desarrollado en diferentes partes que se detallarán separadamente

1. **Entorno de desarrollo:** para la creación de la programación, se ha planteado trabajar con Laravel, que es un framework de código abierto para desarrollar aplicaciones con php. Es un modelo MVC que usa también middleware.
 - a. En la máquina Windows se ha instalado Laravel Homestead, que provee un entorno en máquina virtual con el software necesario ya instalado.
 - b. Se debe asegurar que existe un software para trabajar con máquinas virtuales. En este caso se ha utilizado Virtual Box 5.0.
 - c. Seguidamente se ha de instalar Vagrant, que es una herramienta para la creación y desarrollo de entornos virtualizados que se creó en su inicio para Virtual Box.
 - d. Se ha instalado un software para trabajar con la Terminal de comandos Git Bash en lugar de trabajar con la consola de Windows.
 - e. Y como IDE, se ha instalado 'Sublime Text 3.0' para la programación en php.
 - f. Con las herramientas ya instaladas, se ha procedido a la carga de Homestead, como un box de Vagrant con el comando 'vagrant box add laravel/homestead'. Se pide la plataforma de virtualización con la que se trabajará. En este caso Virtual Box.
 - g. Se crea un directorio que se llamará 'proyectos' para almacenar el software de Laravel.
 - h. Una vez el entorno instalado, se ha debido configurar los ficheros de configuración con el comando 'bash init.sh' en la carpeta /home/user/proyectos/Homestead.
 - i. Este proceso crea el fichero de configuración homestead-yaml'.
 - j. Se crea la llave de configuración que se llamará 'llavehomestead'
 - k. Se mapea una carpeta en el ordenador local (proyectos/Homestead/código) con una carpeta en el vhost (home/vagrant/Code)
 - l. Se define el site con un nombre de dominio irreal (miCompra.com) y se mapea con una carpeta en el vhost(home/vagrant/Code/miSitio/public)
 - m. Para que se reconozca este dominio, se debe modificar la carpeta del sistema de host de Windows, abriendo el bloc de notas en modo administrador: se añade la ip del homestead (192.168.10.10) relacionada con el dominio creado (miCompra.com)

Nota – aunque se ha comentado previamente que, finalmente, no se ha implantado con Laravel, se incluye este punto en la memoria ya que se ha utilizado una parte del desarrollo.

2. **Entorno de desarrollo:** arranque de la máquina virtual
 - a. Se arranca la consola Git Bash
 - b. Se direcciona a la carpeta Home/user/proyectos/Homestead – que es donde se ha creado la carpeta ‘proyectos’ para alojar el Homestead. Desde este último directorio se puede arrancar Vagrant
 - c. Se arranca Vagrant con el comando ‘vagrant up’
 - d. Este proceso arranca la máquina virtual que se creó en Virtual Box
 - e. Se conecta con el servidor virtual con el comando ‘vagrant ssh’.
3. **Entorno de desarrollo:** creación nuevo proyecto Laravel
 - a. Con la máquina iniciada se accede en el servidor al directorio que se creó (y que está definido en el fichero de configuración) ‘Code/ miCompra’
 - b. Se crea una nueva instancia de un proyecto Laravel con el nombre ‘miCompra’
 - c. Se crean todas las carpetas necesarias.
 - d. Se comprueba el funcionamiento accediendo en el navegador a la IP definida para el proyecto (192.168.10.10) – aparece la pantalla de bienvenida
 - e. Se crean todas las carpetas necesarias.
4. **Entorno de desarrollo:** creación de las vistas. Para las páginas de usuario, se ha utilizado Bootstrap, que, a la vez, provee un entorno fácil para una web responsive.
 - a. Se crean los ficheros: /public/css/web.css y /public/js/web.js que utilizaremos en las pantallas de la aplicación.
 - b. Se crea una carpeta /resources/views/web donde se almacenarán las vistas de la aplicación
 - c. Y se crea en /resources/views/layouts/web.blade.php que nos servirá de plantilla. En esta plantilla se incluyen los enlaces siguientes:
 - i. Bootstrap css
 - ii. Bootstrap.js
 - iii. Font Awesome
 - iv. Google api jquery
 - v. A los ficheros web.css y web.js
5. **Entorno de desarrollo:** Gestor visual de administración de la BDD. Se instala en la máquina física el software MySQL Workbench para gestión de la BDD. Se configura con los siguientes parámetros:
 - a. Nombre de conexión: Homestead
 - b. Hostname: 192.168.10.10
 - c. Username: homestead
 - d. Password: ‘secret’
 - e. Puerto: 3306

La estructura de directorios será la siguiente:

- Directorio raíz (public_html en el hosting) - Ubicamos los ficheros de forma jerárquica:
 - o Layouts
 - tfg.php (home)
 - listaCompra.php
 - miListaCompra.php
 - comparar.php
 - confirmar.php
 - o Partes
 - footer.php
 - lateralDer.php
 - laterallzq.php
 - header.php
 - contentHome.php
 - o Programación
 - conexión.php
 - obtener Selección.php
 - seleccionFinal.php
 - o Login
 - desconecta.php (sesión)
 - validar.php (validar login)
 - tfgRegister.php (registrar usuario)
 - tfgLogin.html (pantalla login)
 - tfgRegister.html (pantalla registro de usuario)
 - o Ficheros de trabajo
 - css
 - js
 - fonts
 - pluggins (Bootstrap, jQuery, iCheck, etc)
 - imágenes (jumbotron, thumb, logos, avatares)

8. **Entorno de desarrollo: sistema de autenticación.** Definimos las vistas de login y de registro y se enlazan con la BDD (tabla usuarios):

Se actualiza la visualización del usuario registrado que utiliza la aplicación en el header.

9. **Entorno de pruebas: preparación de las tablas de la BDD.** Se ha realizado una labor de búsqueda en la web para localizar la información.

- a. Se crea una Hoja de cálculo en Excel con cada una de las tablas.
- b. Para el entorno de pruebas se definen los siguientes valores:
 - i. Usuarios: se definen sólo dos usuarios.
 - ii. Supermercados: se definen los 8 supermercados planificados
 - iii. Sucursales: se definen (3) sucursales de Caprabo y (2) de Condis. Todos ellos en el mismo distrito postal. En Google Maps se consiguen sus latitudes y longitudes.
 - iv. Marcas: se definen (57) marcas relacionadas con las categorías elegidas.
 - v. Categorías: se definen las 9 categorías principales
 - vi. Sub-categorías: se definen las (63) sub-categorías principales

- vii. Sub-divisiones: Se definen las sub-divisiones de las subcategorías 'aceites', 'aceitunas', 'agua' y 'refrescos'.
- viii. Productos: Se definen todos los productos que contendrá la aplicación. Es la segunda tabla más extensa en número de registros.
- c. Se localizan los logos de los supermercados y se guardan en un subdirectorío. Con Photoshop se guardan con la misma extensión y el mismo tamaño.
- d. Se localizan las imágenes de 'n' productos. Se almacenan en un subdirectorío propio estructurado por categoría y marca para su mejor localización.

11. APIs utilizadas

Se relaciona a continuación, una descripción resumida de las APIs utilizadas:

(1) LARAVEL 5.4 - En el proceso de desarrollo se ha utilizado el framework Laravel, versión 5.4.

Es un framework tipo MVC (Modelo Vista Controlador), que también utiliza middlewares.

Utiliza lenguaje PHP, como se ha detallado en el apartado [7] de 'Arquitectura de la Información'

(2) Font Awesome: fuentes e iconos vectoriales – Versión 4.7.0 (<http://fontawesome.io/>)

(3) Ionicons – iconos Versión 2.01 (<http://ionicons.com/>)

(4) Bootstrap (Versión 3) – framework para diseño responsive (<http://getbootstrap.com/>)

(5) ADMIN LTE2 – Plantilla template Bootstrap

(<https://colorlib.com/wp/free-bootstrap-admin-dashboard-templates/>)

(6) PIXABAY.COM – Repositorio de imágenes y videos libres de derechos de autor. Se han utilizado imágenes para el Jumbotron de la página de inicio (formato jpg de tamaño modificado con Photoshop a 1920x500) y el logo de la aplicación (en formato png). (<https://pixabay.com/es/>)

(7) GOOGLE FONTS – Tipografías (hasta 818 diferentes) (<https://fonts.google.com/>)

(8) GOOGLE MAPS – Generador de mapas versión 3

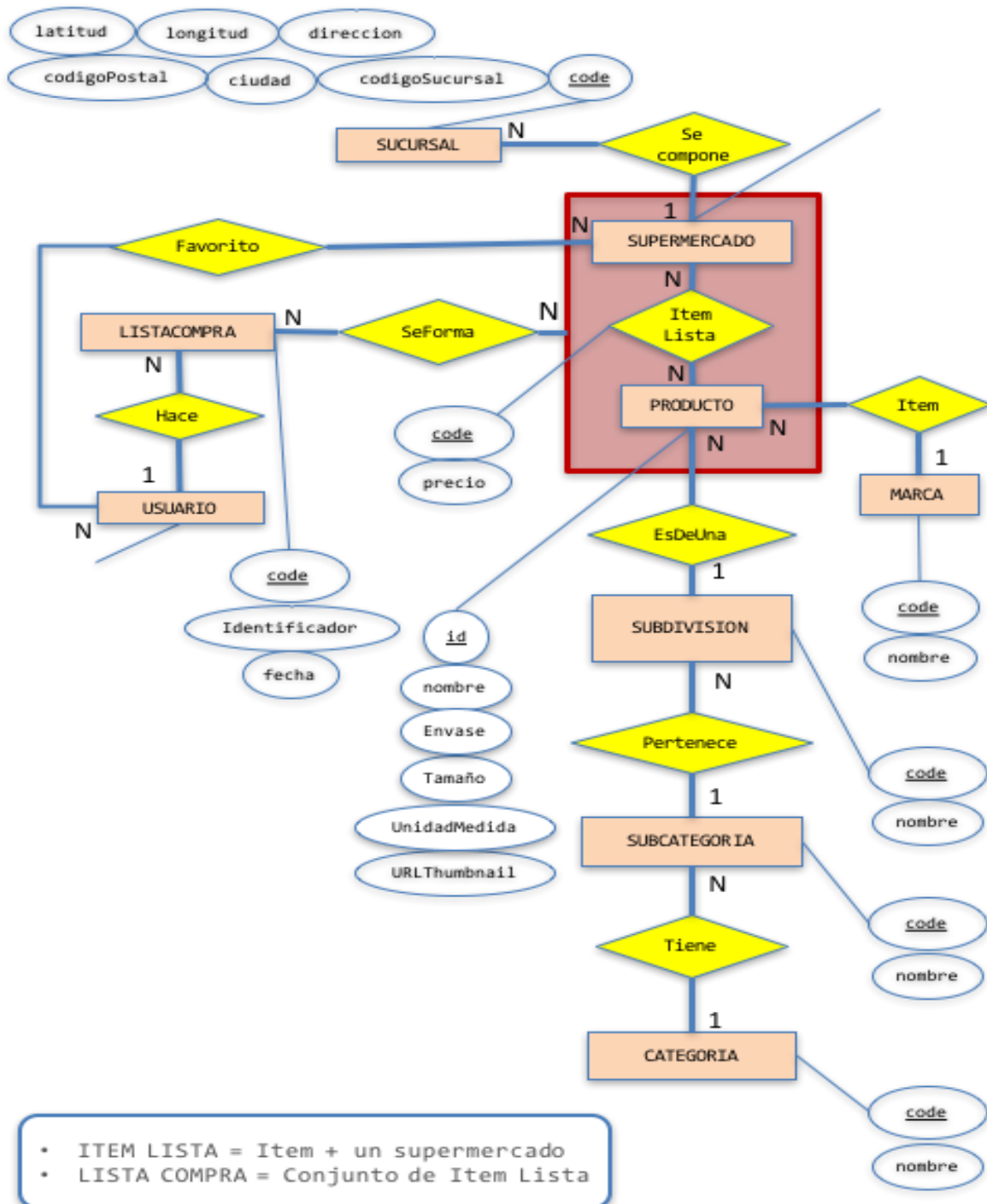
(<https://developers.google.com/maps/documentation/javascript/?hl=es-419>)

(9) TCPDF – Generador open source de páginas PDF (<https://tcpdf.org/>)

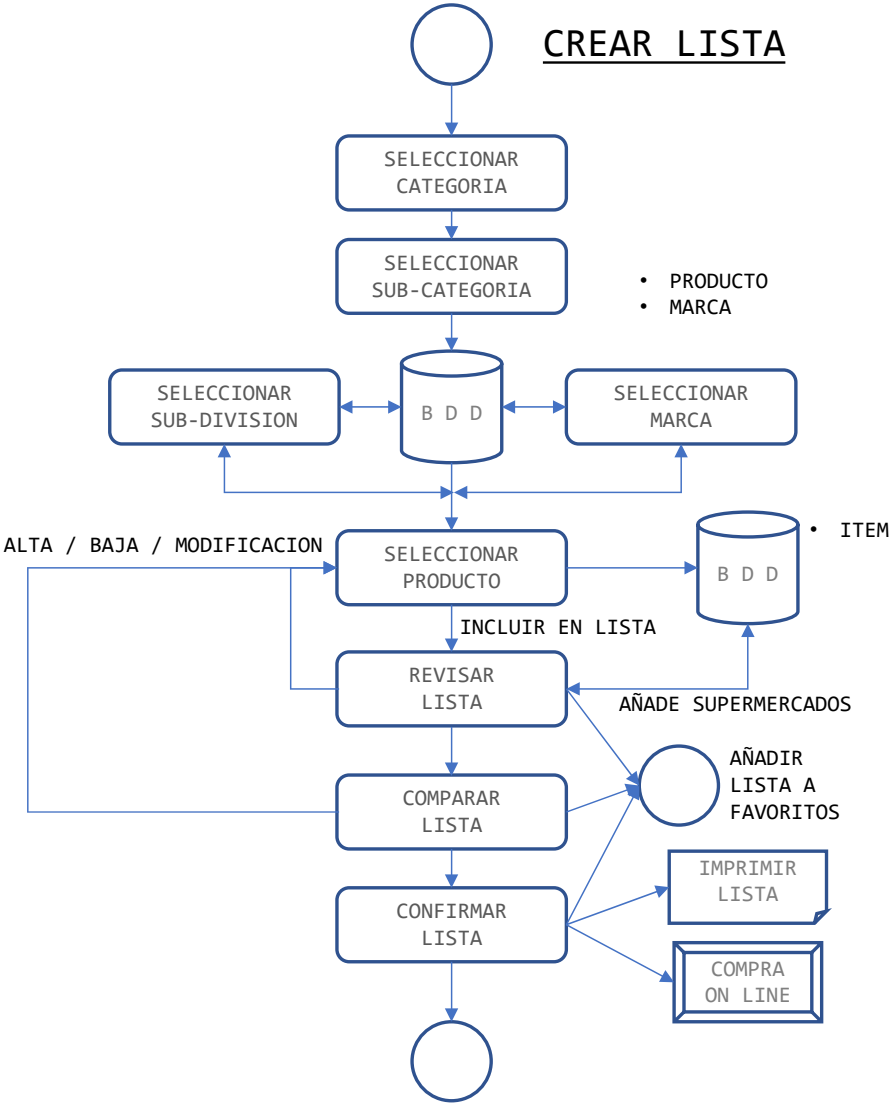
12. Diagramas.

12.1 Diagrama Entidad / Relación

- Las listas de compra realizadas por usuarios no registrados no serán guardadas.
- No se almacenarán listas de compra de más de 2 años de antigüedad.
- Un 'ítem' (producto + marca) siempre tiene el mismo precio en todas las sucursales del mismo establecimiento



12.2 Flow Chart – Creación de una Lista de Compra



12.3 Casos de uso

Un caso de uso es una descripción de un conjunto de secuencias de acciones que ejecuta un sistema y que recoge un contrato entre los stakeholders y el propio sistema, mediante su comportamiento observable.

Siempre hay un stakeholder que es especial, que es quien quiere usar el sistema para satisfacer un objetivo concreto. El caso de uso nos describe cuál es el comportamiento observable del sistema durante esta ejecución del caso de uso, de manera que se pueda asegurar que se cumple el objetivo del actor principal teniendo en cuenta los intereses del resto de los stakeholders

CU001: Registro en la aplicación.

Actor principal: usuario

Ámbito: aplicación

Nivel de objetivo: usuario

Stakeholders e intereses:

- el usuario desea entrar en la aplicación por lo que debe efectuar login

Precondición:

- se ha accedido a la pantalla de inicio

Garantías mínimas: el sistema ofrecerá al usuario acceso a la página de login.

Garantías en caso de éxito: el sistema permitirá acceso a la página de selección y creación de una lista de compra.

Escenario principal de éxito:

1. El usuario indica al sistema que desea crear una lista nueva en la página de inicio por una de las siguientes 3 opciones:
 - a. Con click en logo de la aplicación
 - b. Con click en botón de acceso
 - c. Con selección de subcategorías
2. El sistema direcciona la aplicación a la página de creación de lista de compra.
3. Si no hay sesión abierta con el usuario, se abre la pantalla de login, donde el usuario puede registrarse.
4. El sistema valida que exista la combinación de usuario y contraseña en la bdd.
5. Si no existe, el sistema le permite crear un usuario nuevo en el sistema en la vista de Registro
6. El sistema actualiza la tabla de usuarios y le permite el acceso a la pantalla de selección de productos donde puede efectuar la selección de los productos que busca.

CU002: Crear una lista de compra nueva.

Actor principal: usuario

Ámbito: aplicación

Nivel de objetivo: usuario

Stakeholders e intereses:

- el usuario desea realizar una lista de compra nueva

Precondición:

- el usuario está registrado,

Garantías mínimas: el sistema ofrecerá al usuario acceso a la página de selección de productos donde se mostrarán 'n' productos obtenidos aleatoriamente de la base de datos.

Garantías en caso de éxito: el sistema solicitará confirmación al usuario para efectuar la comparación de precios. No se registrará en el sistema hasta que no exista esta confirmación.

Escenario principal de éxito:

1. El usuario indica al sistema que desea crear una lista nueva en la página de inicio.
2. El sistema le presenta la pantalla de selección de productos donde ya se muestran 'n' productos elegidos aleatoriamente.
3. El usuario selecciona la categoría
4. El sistema le muestra las subcategorías asociadas a esa categoría con un menú desplegable.
5. El usuario selecciona la subcategoría
6. El sistema le muestra las subdivisiones asociadas a esa subcategoría en el espacio reservado al efecto en forma de checkbox.
7. El sistema le muestra las marcas asociadas a la subcategoría seleccionada en el espacio reservado al efecto en forma de checkbox.
8. El usuario, si lo desea, selecciona una subdivisión. En este caso las marcas posibles se actualizan consecuentemente.
9. En función de las acciones realizadas en los puntos 3 – 8, los productos seleccionados de la BDD se muestran en su espacio reservado, y le muestra la selección que está realizando, así como el número total de productos seleccionados.
10. El sistema muestra en los puntos 3 – 8, la cantidad de posibles selecciones que se dispone en la BDD.
11. El usuario selecciona el producto que desea y le indica al sistema que desea incluirlo en la lista
12. El sistema le indica cuantos productos ya tiene seleccionados.
13. El usuario decide revisar la lista
14. El sistema lo direcciona a la página de revisión
15. El usuario revisa la lista

- 16. El usuario decide finalizar la lista y efectuar la comparación
- 17. El sistema le direcciona a la página de comparación

Extensiones:

- 15.a El usuario desea incluir un ítem adicional
El sistema lo direcciona a la página de selección.
- 15.b El usuario decide eliminar un ítem de la lista
El usuario lo selecciona y lo elimina
El sistema lo elimina de la lista temporal de trabajo.
- 16.a El usuario decide incluir esta lista en sus listas Favoritas
Previamente el usuario debe de estar registrado.
El sistema almacena esa lista en las favoritas de selecciona el producto que desea
y le indica al sistema que desea incluirlo en la lista

CU003: Comparar una lista de compra.

Actor principal: usuario

Ámbito: aplicación

Nivel de objetivo: usuario

Stakeholders e intereses:

- el usuario desea comparar los precios de su lista de compra

Precondición:

- el usuario puede o no, estar registrado

Garantías mínimas: el sistema ofrecerá al usuario acceso a la página de comparación de la lista y registrará la lista de compra.

Garantías en caso de éxito: el sistema solicitará confirmación al usuario para efectuar la confirmación de la lista para su gestión posterior.

Escenario principal de éxito:

1. El sistema muestra una tabla con los productos de la lista y los precios de los supermercados.
2. El sistema no mostrará precio si el producto no existe en ese supermercado en concreto (puede suceder con marcas blancas, por ejemplo).
3. El usuario revisa los totales de la lista por establecimiento y selecciona los supermercados que desee pulsando en el cuadro de texto
4. El sistema los identifica e identifica con un código de color los precios más altos y más bajos.
5. El usuario revisa los precios de los productos
6. El usuario confirma la elección de los supermercados, una vez efectuada la revisión.

Extensiones:

- 4.a Si el usuario está registrado, el sistema marca sus establecimientos favoritos.
- 5.a El usuario desea incluir un ítem adicional
El sistema lo direcciona a la página de selección.
Una vez seleccionado el ítem adicional, el sistema lo direcciona a la página de origen
El sistema actualiza la lista temporal
- 5.b El usuario decide eliminar un ítem de la lista
El sistema lo direcciona a la página de revisión.
El usuario decide las acciones a realizar
El sistema actualiza la lista temporal
- 5.c El usuario decide modificar la lista entera
El usuario sistema lo direcciona a la página de revisión
El usuario decide las acciones a realizar

- 16.a El sistema lo direcciona a la página de origen, una vez finalizada la acción.
El usuario decide incluir esta lista en sus listas Favoritas
Previamente el usuario debe de estar registrado.
El sistema almacena esa lista en las favoritas de selecciona el producto que desea y le indica al sistema que desea incluirlo en la lista

CU004: Confirmar la lista de compra.

Actor principal: usuario

Ámbito: aplicación

Nivel de objetivo: usuario

Stakeholders e intereses:

- el usuario desea confirmar la lista de compra en los establecimientos elegidos

Precondición:

- el usuario puede o no, estar registrado

Garantías mínimas: el sistema ofrecerá al usuario acceso a la impresión de la lista o el acceso a la compra on-line.

Garantías en caso de éxito: el sistema imprimirá la lista de la compra o el acceso a la gestión on-line.

Escenario principal de éxito:

1. El sistema muestra una tabla con los productos de la lista y los precios de los supermercados elegidos.
2. El usuario indica el distrito postal de su ubicación
3. El sistema muestra un mapa con las sucursales de los establecimientos elegidos situados en el mismo distrito postal.
4. El usuario revisa los totales de la lista por establecimiento y selecciona los productos que desea comprar en cada supermercado.
5. El sistema los identifica e identifica con un código de color y actualiza los totales por selección por establecimiento.
6. El usuario revisa la selección
7. El usuario imprime la lista de la compra.
8. El sistema registra la lista de la compra

Extensiones:

- 2.a Si el usuario está registrado, el sistema utiliza de inicio el distrito postal del usuario.
El usuario registrado puede también modificarlo
- 6a El usuario desea incluir un ítem adicional
El sistema lo direcciona a la página de selección.
Una vez seleccionado el ítem adicional, el sistema lo direcciona a la página de origen
El sistema actualiza la lista temporal
- 6.b El usuario decide eliminar un ítem de la lista
El sistema lo direcciona a la página de revisión.

- El usuario decide las acciones a realizar

El sistema actualiza la lista temporal

6.c El usuario decide modificar la lista entera

El usuario sistema lo direcciona a la página de revisión

El usuario decide las acciones a realizar

El sistema lo direcciona a la página de origen, una vez finalizada la acción.
- 7.a El usuario decide incluir esta lista en sus listas Favoritas

Previamente el usuario debe de estar registrado.

El sistema almacena esa lista en las favoritas de selecciona el producto que desea y le indica al sistema que desea incluirlo en la lista
- 7.b El usuario puede decidir acceder a la compra on-line para uno o varios de los supermercados elegidos.

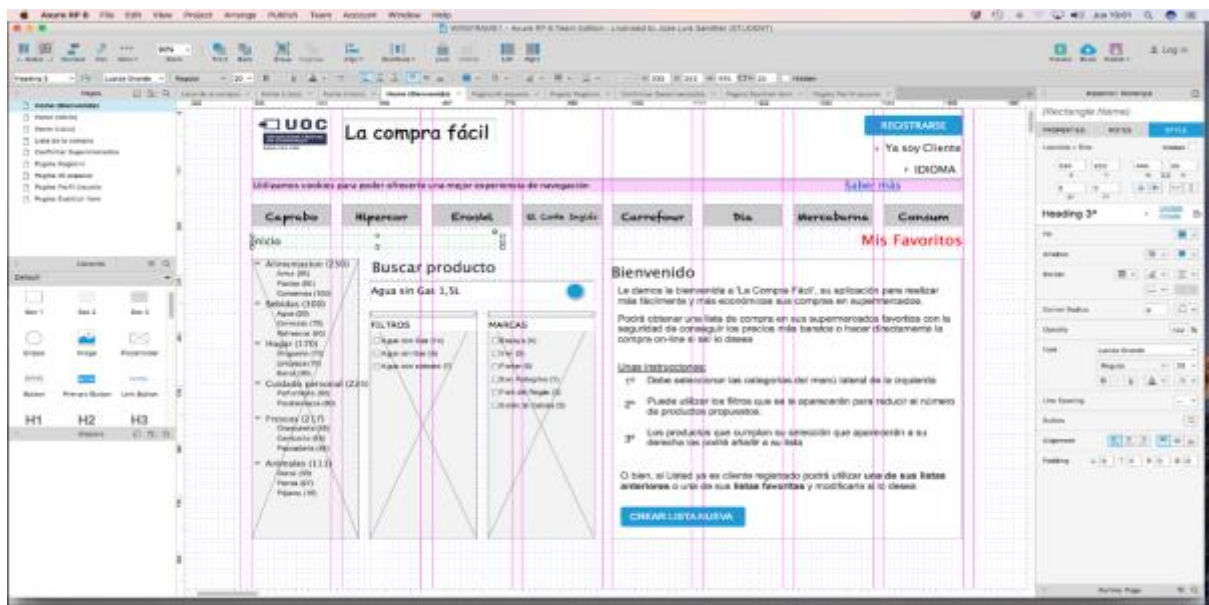
El sistema de acceso a la página web del establecimiento elegido.
- 8.a Si el usuario está registrado, el sistema actualiza el histórico.

13. Prototipos

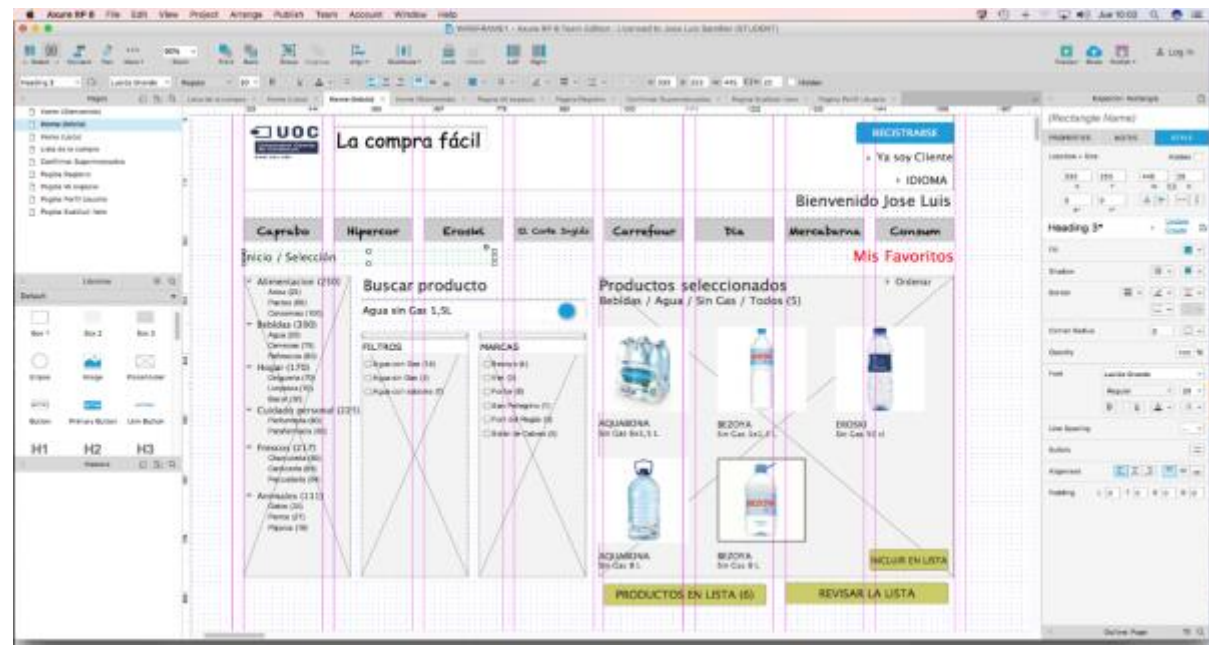
13.1 Lo-Fi - Hi-Fi

El prototipado se ha realizado utilizando 'Axure RP8' del que se dispone una licencia de estudiante. Se anexará a esta memoria el fichero *.rp. Y se anexa copias de las pantallas, a continuación.

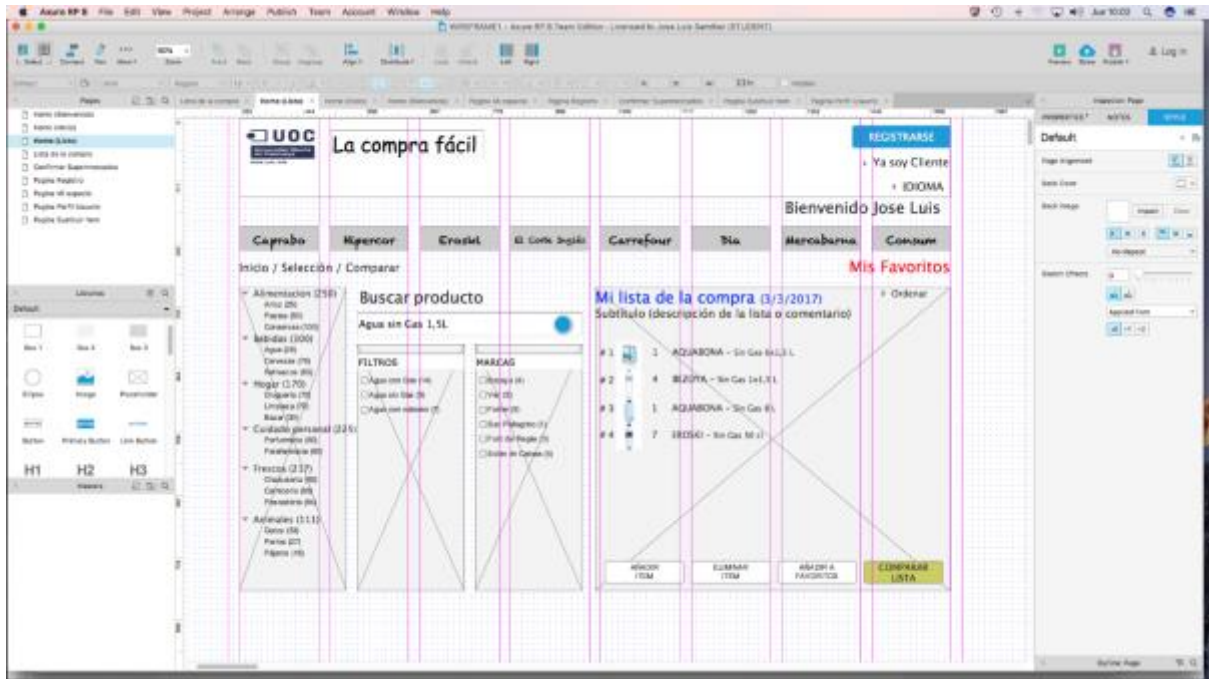
Pantalla Home / Bienvenida



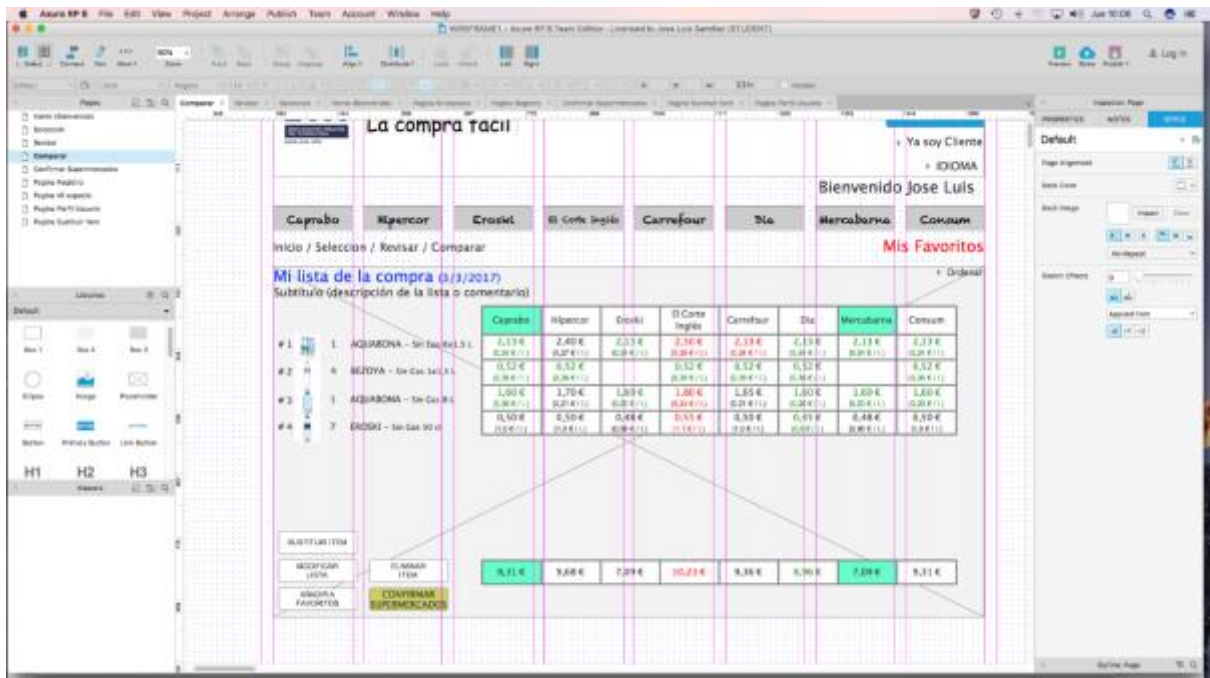
Pantalla Home / Selección



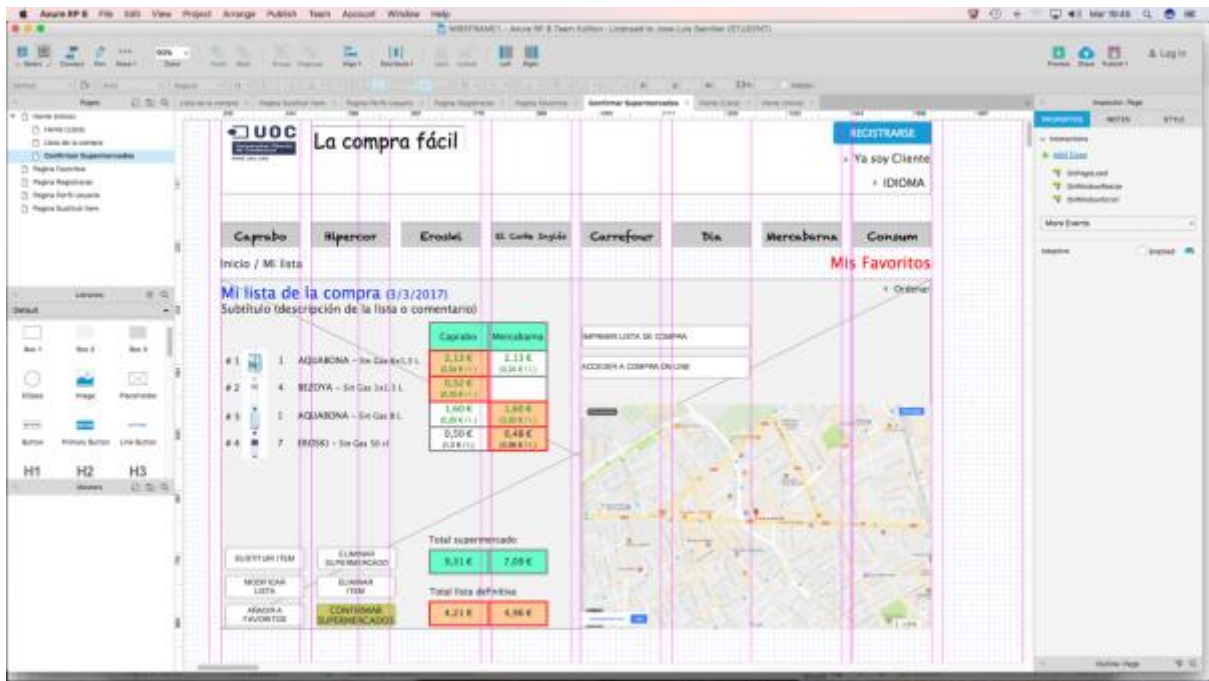
Pantalla Home / Selección / Revisar



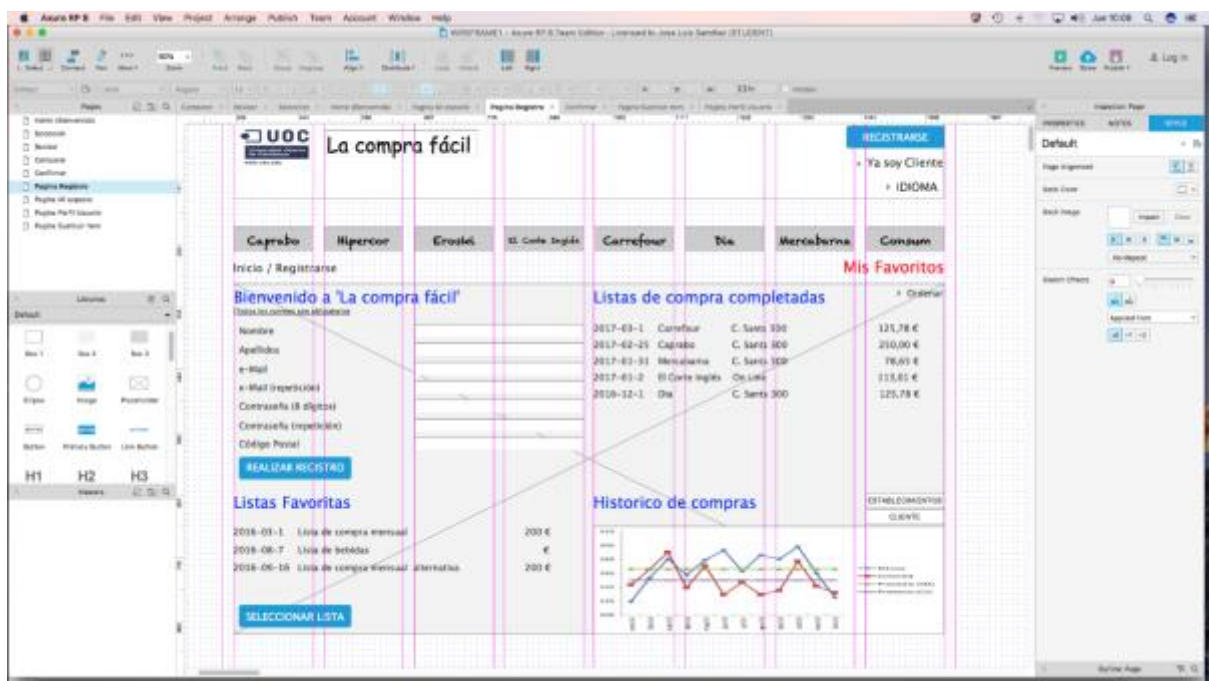
Pantalla Home / Selección / Revisar / Comparar



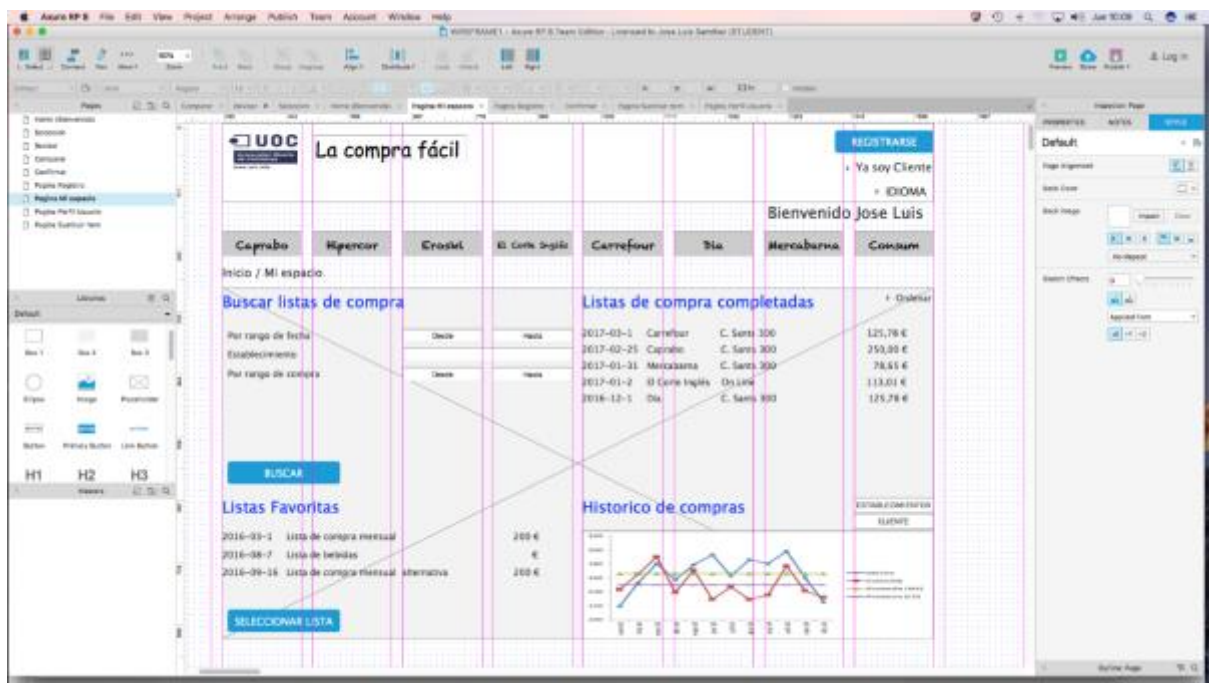
Pantalla Home / Selección / Revisar / Comparar / Confirmar



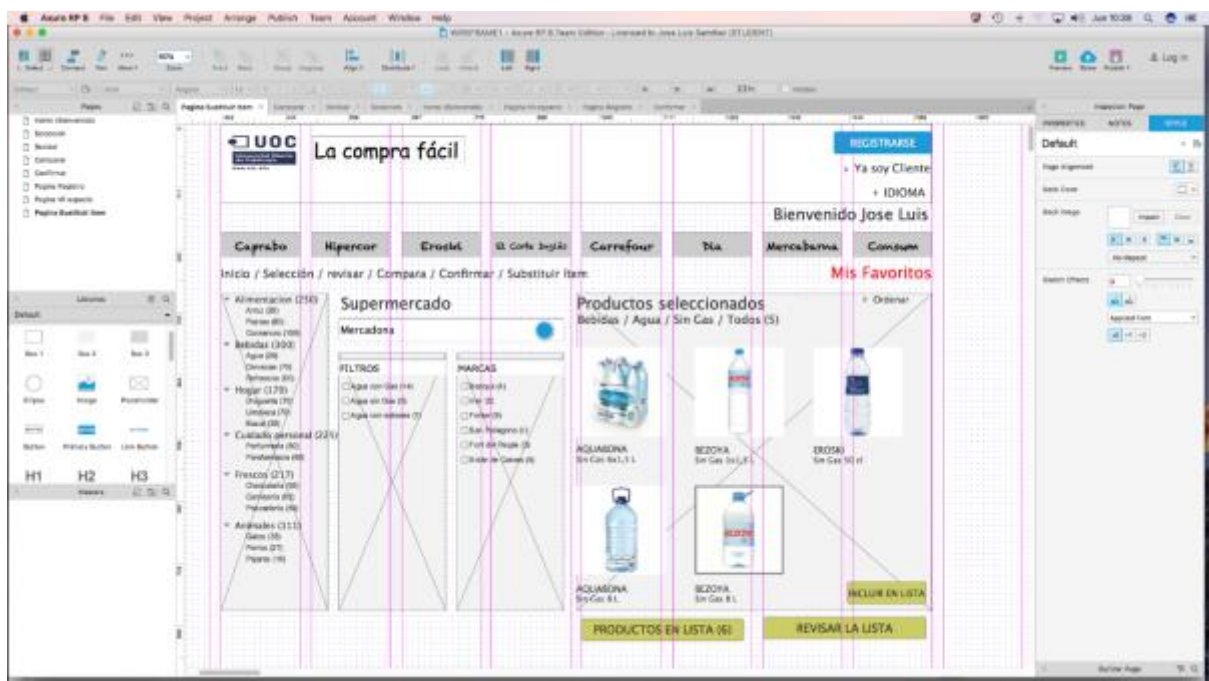
Pantalla Home / Registro



Pantalla Home / Mi espacio



Pantalla Home / Sustituir ítem



14. Perfiles de usuario

Para el diseño de la aplicación, se establecen unas fichas de usuario:

- Eva / Ramón
 - Edad 30 a 40 años
 - Profesión liberal
 - Descripción de la persona: casad@ con 2 hijos. Vive cerca de Barcelona. Trabaja en Barcelona. Dispone de poco tiempo libre entre semana. Tiene un día establecido para la compra general, pero suele ir entre semana para temas puntuales. Disponen de conexión a Internet en casa y tienen un dominio medio/alto de su Smartphone. Realizan la compra general cada 15 días.
 - Escenario: durante la semana va accediendo a una aplicación que le recomendaron donde va registrando los productos que necesita comprar. Incluso dispone de una lista tipo de compra quincenal. Cuando necesita efectuar la compra, la aplicación le sugiere los establecimientos más baratos y cercanos donde realizarla. Puede incluso dividirla en varias partes para efectuar una compra física y otra on-line. También puede compartirla con su marido/mujer y acceder desde su Smartphone. Incluso puede modificar la lista en cualquier momento y lugar si se acuerda de un producto concreto.

- Luis
 - Edad 67 años
 - Jubilado
 - Está casado y ya no tiene hijos a su cargo. Vive en Barcelona. Tiene un conocimiento básico de ordenadores, pero está adaptado a su uso. Dispone de conexión a Internet en el domicilio y de un Smartphone básico
 - Esporádicamente va entrando en la aplicación y va apuntando los productos que necesita. Como dispone de tiempo, elige los supermercados más barato, aunque tenga que ir a 2 o 3 establecimientos diferentes. El sistema le imprime la lista por comercio o incluso puede consultarla en su Smartphone

Ambos son usuarios focales.

Para la realización de la aplicación, se establecen tres tipos de usuarios:

- 1) Usuario focal esporádico - Hace uso del sistema
 - Creará listas de compra y las modificará
 - Comparará los precios de los productos y elegirá los supermercados
 - Imprimirá la lista
 - Se conectará on-line

- 2) Usuario focal habitual – Hace uso del sistema y, además
 - Se da de alta como usuario registrado en la aplicación
 - Crea y gestiona sus listas favoritas
 - Accede a sus históricos

- 3) Usuario focal Administrador
 - Tiene el acceso como usuario a la aplicación
 - Crea y mantiene la BDD general.

15. Usabilidad

Se ha desarrollado un diseño centrado en el usuario desde el comienzo.

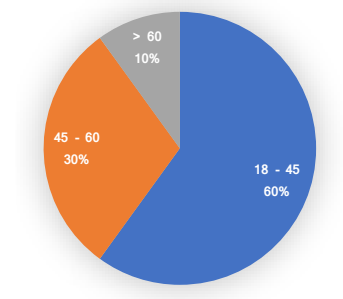
En el diseño de la aplicación se ha intentado, por una parte, mantener una coherencia en la distribución de los elementos. Y por otra, conseguir una sencillez en cuanto a la búsqueda de los productos.

Esto es complicado ya que la cantidad de ellos es muy elevada. Se necesita una categorización que debe ser sencilla, para no obligar al usuario a muchas decisiones, pero, también, que permita reducir al máximo la cantidad de productos mostrados en la última selección.

El usuario de esta aplicación será mayoritariamente joven. Sobre todo, cuando se termine la aplicación móvil.

Sin embargo, en el diseño de esta aplicación, se tendrá en cuenta al usuario 'senior'. Por ello,

- El tamaño de letra se mantendrá por encima de 12.
- La tipografía será 'sans serif'.
- Evitaremos el uso de menús desplegados, con la sola excepción de la identificación para el acceso.
- La navegación será principalmente secuencial (ayudado también por el tipo de aplicación). Y se utilizará un 'breadcrumb' para mostrar en qué parte de la aplicación está el usuario.



Aunque no sea un punto que preocupe al usuario joven, no se pedirá registrarse para usar la aplicación. Solo se solicitará para el uso del concepto 'favoritos' o 'mi espacio'. En ningún caso se solicitará DNI, sino un registro por 'e-mail + contraseña'.

Para facilidad y rapidez de carga, las fotografías no serán ampliables ya que también se considera suficiente el tamaño mostrado para la identificación del producto, ni se ofrecerá información de detalle del producto, más de lo estrictamente necesario para su correcta elección.

Se han presentado en el punto anterior una descripción de los usuarios tipo que podrán hacer uso de la aplicación.

Se ha tenido en cuenta en el diseño el uso de una retícula para situar los elementos en la página y también para facilitar la adaptación al tamaño de la pantalla. Como ya se ha expresado, será una web 'responsive'.

Para su realización se ha utilizado el software Axure RP8, que permite el uso de retículas y la visualización en el navegador de lo que se está diseñando.

La información está estructurada para un seguimiento de 'izquierda a derecha': En las pantallas de selección, se va filtrando desde la izquierda. Y se propone la selección a la derecha.

Un punto importante en esta aplicación es el de la selección de los productos deseados ya que la BDD de productos es enorme debido principalmente a su variabilidad (por marcas y por supermercados).

Para ello, se dispondrá de dos métodos principales:

- 1) Utilizando una clasificación jerárquica, en base a categorías, subcategorías y filtros. Como ejemplo, si deseamos elegir una botella de agua, seleccionaremos
 - Categoría – Bebidas
 - Subcategoría – Agua
 - Filtro 1 – ‘Sin gas’
 - Filtro 2 – Marca de producto
- 2) Utilizando una denominación más ‘natural’ para el usuario. En un cuadro de texto, se introducirá lo que desea, sin la necesidad de saber cómo está estructurada la BDD. Siguiendo el ejemplo anterior, se escribirá. “Agua sin gas de 1,5 l”. La aplicación debe conocer lo que solicita el usuario. Para ello:
 - Descompone el texto en una serie de palabras que se utilizarán como ‘key items’
 - En este caso: ‘agua’ + ‘sin gas’ + ‘1,5’
 - Con esta información se accederá a las tablas de ‘producto’ o ‘marca’
 - El algoritmo de búsqueda es un punto importante de la aplicación.
- 3) En una segunda fase, se habilitará la selección por código de barras o código QR. Incluso puede plantearse la selección por voz.

Se tendrá en cuenta la prevención de errores en la introducción de datos:

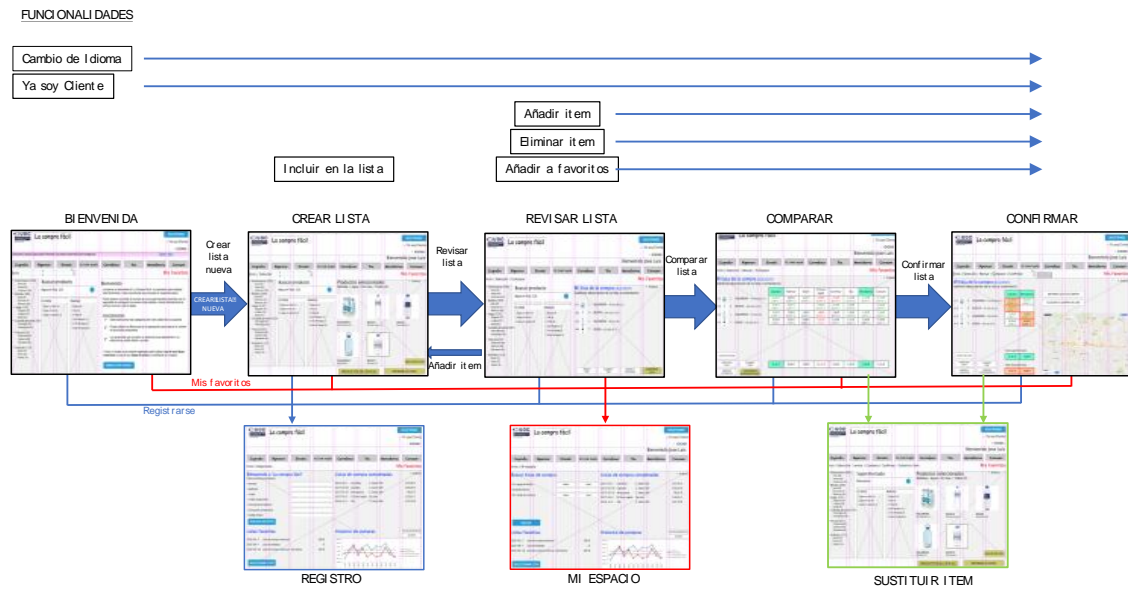
- En campos numéricos, de e-mail o de fecha.
- El sistema irá actualizando la información solicitada según se vayan introduciendo más filtros.

Con respecto a la gama cromática se ha decidido la utilización de las siguientes combinaciones:

- Por impacto visual, las combinaciones más favorables son:
 - Negro sobre Blanco.
 - Negro sobre Amarillo – aunque éste no se utilizará debido a que el color amarillo transmite unas sensaciones que se considera no apropiadas (adolescencia, celos, envidia, egoísmo...).
 - Negro sobre azul muy claro – para dar otra posibilidad a la primera elección y debido a que la segunda se ha descartado.
- Por sensación de color se utilizarán los colores siguientes, que aportan:
 - Negro – seriedad y nobleza.

- Blanco – estabilidad, calma, armonía.
- Azul – seriedad, nobleza, elegancia.

Se muestra a continuación un mapa del sitio en el que se refleja también la navegación



16. Seguridad

Se van revisar dos aspectos referentes a la seguridad:

1. Servidor:
 - a. Acceso: asegurado por la empresa que gestiona el hosting. Se dispone de un usuario y clave que se irá modificando periódicamente.
 - b. Copias de seguridad cada 4 horas gestionadas por el hosting
 - c. Análisis de los logs. Reactivamente cada semana
 - d. Al tratarse de un servidor de alojamiento compartido:
 - i. Configuramos 'open_basedir' – limita las operaciones con ficheros al directorio especificado
 - ii. Se protegen las variables de sesión (sesión_set_saver_handler())
 - iii. Se cifran los datos de la carpeta Temp
2. Bases de datos
 - a. Copias de seguridad cada 4 horas gestionadas por el hosting
 - b. En los inputs de formulario (login y registro) se limitará el número de caracteres para impedir que el input exceda las dimensiones de la tabla
 - c. La contraseña se guardará codificada en la tabla de usuarios con la función
`$hashRegister= password_hash ($pwd, PASSWORD_DEFAULT);`
 - d. Se utiliza:
 - i. Función htmlentities()
 - ii. Librería HTML Purifier (pendiente de implementar)
3. Formularios
 - a. Implementar un código captcha (pendiente de desarrollar)

17. Test

Se realizan diferentes tipos de pruebas:

1. Usabilidad
 - Por usuarios expertos – en la fase de desarrollo y en la fase ya final de pre entrega
 - Por el propio desarrollador
2. Seguridad
 - Por el propio administrador- se realiza una lista de posibles acciones que se van probando

18. Versiones de la aplicación/servicio

1. Primera versión de desarrollo – contiene las funcionalidades sin que se relacionen entre ellas:
 - a. Bases de datos
 - b. Layout de las páginas con datos estáticos
 - c. Diseño
 - d. Integración de plantillas
2. Segunda versión (Beta) – se empiezan a integrar las funcionalidades por fases.
 - a. Dominio y hosting
 - b. Pruebas de acceso
 - c. Modificación de las BDD
 - d. Login y Registro
 - e. Generación dinámica de selección
 - f. Creación de listas de compra
 - g. Enlaces
3. Tercera versión (primera entrega, versión 1.0)

19. Requisitos técnicos

Usuario

- Se requiere un nivel de conocimientos básicos
- En cuanto al hardware, y en esta primera fase, con un ordenador conectado a la web y a una impresora será suficiente.

Servidor

- Se necesita un hosting dedicado (en este caso Webempresa).
- Debe poder manejar PHP y MySQL
- En nuestro caso, el hosting trabaja con Linux
- Se dispone de:
 - o Certificado SSL
 - o Discos SSD (3Gb + 9 de backup)
 - o Copias de seguridad cada 4 horas
 - o Sin límite al número de visitas
 - o Soporte 24 horas 365 días
 - o 120 Gb de transferencia mensual
 - o Aunque sea servidor compartido, se dispone de cuentas aisladas (CloudLinux, CageFS)
 - o Asistencia en castellano

20. Instrucciones de uso

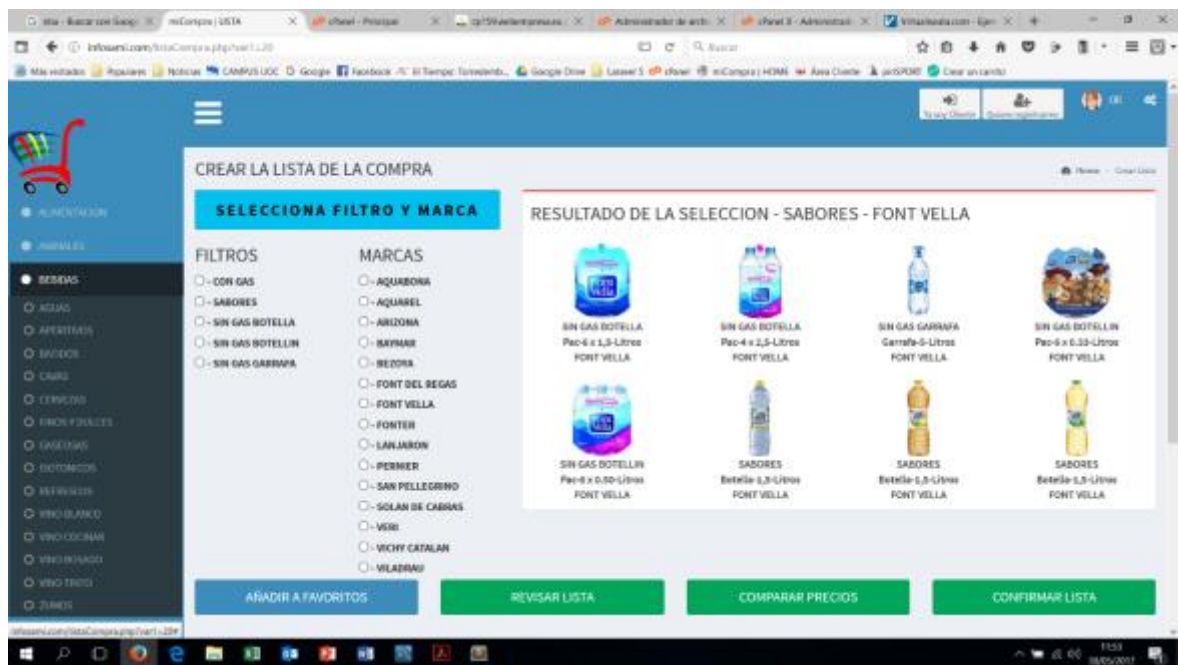
Para utilizar la aplicación, se accederá desde un navegador introduciendo la siguiente dirección: "infosami.com/tfg.php, que nos direccionará a la página de inicio.



Cuando se quiera acceder al contenido, primero deberemos registrarnos en la aplicación.

Si el usuario ya está creado, con el login será suficiente. En caso contrario, se deberá primero crear el usuario y después acceder.

Una vez efectuado el registro, se podrá iniciar la lista

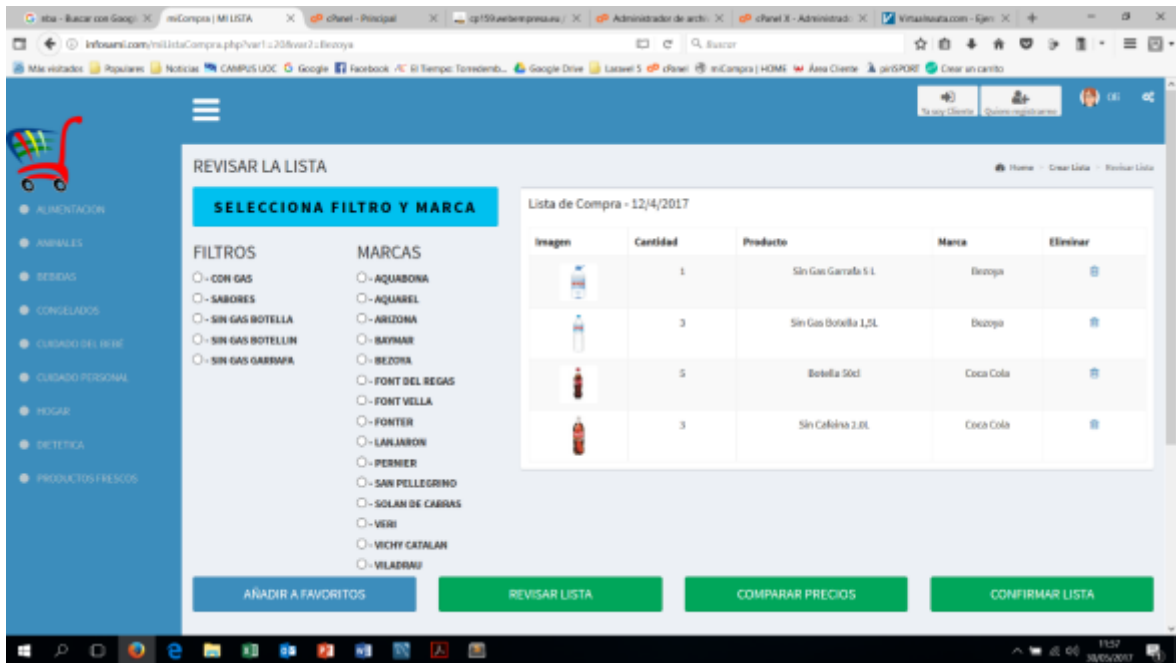


Deberemos elegir los siguientes conceptos:

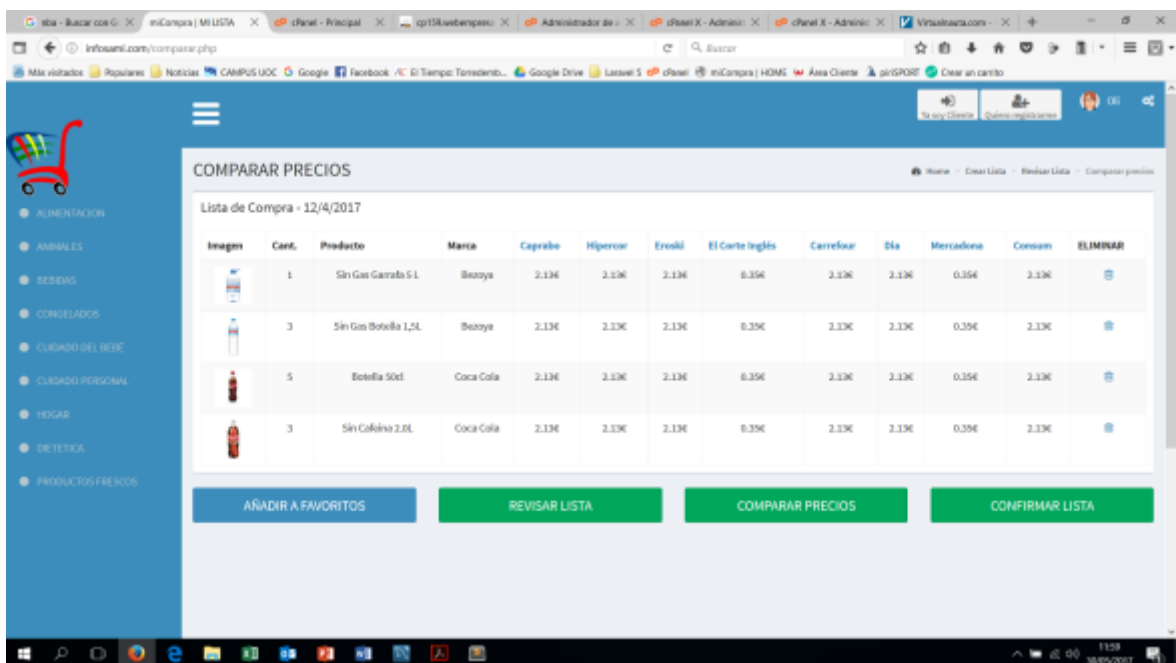
- Categorías
- Subcategorías
- Filtro
- Marca

Lo que nos generará una relación de productos susceptibles de añadir a la lista.

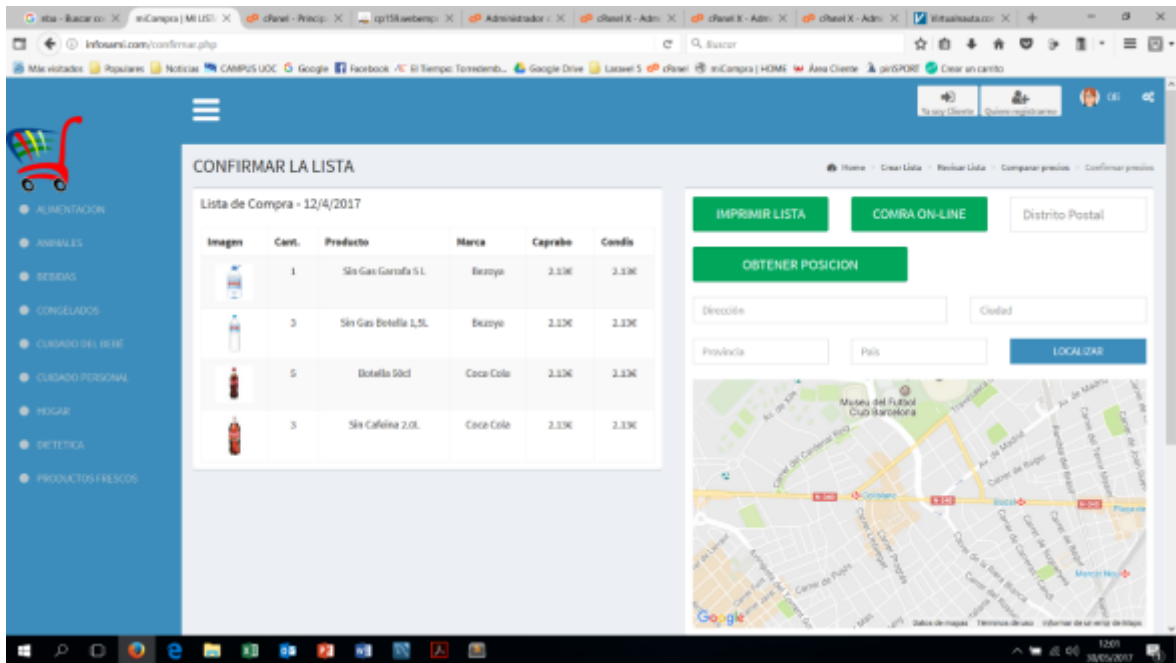
Los productos se irán añadiendo a la lista, clicando sobre ellos. Una vez finalizada la selección, se submitirá un proceso para la revisión de dicha lista



Una vez la lista esté revisada, ejecutaremos un proceso que nos comprobará los precios de los productos en nuestros supermercados favoritos.



En esta vista, podremos elegir los dos supermercados que resulten mas interesantes y se confirmará la lista.



Finalmente, se podrá elegir entre la impresión de una lista en pdf o el acceso a la url de la compra on line del supermercado elegido.

Visualizando en el mapa los centros más cercanos.

21. Bugs

Los principales errores detectados han sido de enlaces a las bibliotecas correspondientes y de funcionamiento general del framework, debido a su novedad.

Agravado por la dificultad de la instalación de Laravel en el hosting, se tomó la decisión de realizar la aplicación sin este framework y desarrollarlo todo a medida, con el inevitable retraso en la planificación que ha afectado a las funcionalidades finalmente realizadas.

Otros tipos de errores han sido de formato y de funcionamiento, por la lista de ficheros .CSS y .js utilizados, Y por la instalación de la plantilla Admin.Lte.

La BDD se ha debido ajustar a las especificaciones finales.

Para considerar que todos los errores (o mejoras) estén cerrados, se deberá ajustar los puntos siguientes:

- Avisos de error en los registros duplicados.
- Ajustes en algún botón en tamaño smartphone
- Mejoras en la implementación de AJAX en todos los puntos susceptibles de ser utilizados
- Los aspectos de seguridad comentados en el punto 16 no implementados

22. Proyección a futuro

Información, predicciones y sugerencias acerca de ampliaciones a futuro del trabajo, y/o lista de mejoras a realizar en hipotéticas futuras versiones del servicio/aplicación.

A continuación, relacionamos las siguientes posibilidades de ampliación que dividiremos en tres grupos:

1. Contempladas en este proyecto, pero no desarrolladas.
 - 1.1. Posibilidad de cambiar el idioma de la aplicación

2. No contempladas, pero a incluir.
 - 2.1. Efectuar selección por código de barras.
 - 2.2. Efectuar selección por código QR
 - 2.3. Posibilidad de poder acceder desde esta página a los servicios de atención al cliente de los diferentes establecimientos
 - 2.4. Acceso a redes sociales: Facebook.
 - 2.5. Parametrizar el número de establecimientos. Por defecto, de inicio, son 8. Conforme crezca la BDD con nuevos establecimientos, se debería poder elegir.
 - 2.6. Utilizar la geolocalización.

3. No contempladas, a estudiar.
 - 3.1. Contemplar la usabilidad para usuarios con visión reducida.
 - 3.2. Realizar App para smartphones que recoja la información de la aplicación web

23. Presupuesto

El presupuesto para la realización de este proyecto lo dividimos en varias partes:

1. Equipo humano: lo dividimos en varias partes
 - a. Planificación del proyecto: 10 horas
 - b. Formación en Laravel. 25 horas
 - c. Diseño: 30 horas
 - d. Programación: 50 horas
 - e. Documentación y generación de la BDD: 25 horas
 - f. Entrega final, pruebas 25 horas

Para calcular el coste, aunque realizado para un proyecto Final de Grado, lo realizaremos a una tarifa de 30 € por hora. Lo que nos da un coste de 4.950 €

2. Infraestructura
 - a. No se considera un local para la actividad
 - b. NI, por tanto, mobiliario ni infraestructura básica.
 - c. No se considera un coste de amortización de equipos informáticos
 - d. Tampoco se considera un coste de mantenimiento de la aplicación
 - e. No considerado el coste de mantenimiento de la BDD con los datos actualizados
3. Parte técnica
 - a. La parte del software propietario dedicado la considero a coste cero ya que se han utilizado licencias financiadas al ser de uso académico o de prueba.
 - b. La aplicación está alojada en el hosting 'Webemprea' con un coste anual de 95,83 €
 - c. Por tanto, no hay gastos de energía, alquileres ni depreciaciones
 - d. Las tareas de mantenimiento del hosting están incluidas.
 - e. No consideramos gastos de conexiones a Internet
 - f. El equipo hardware informático utilizado se considera también a coste cero ya que no se han tenido que comprar equipos nuevos.

24. Análisis de mercado

Esta aplicación está pensada para crear una lista de compra que se validará con los establecimientos de referencia para obtener sus precios de venta y, de esta manera, conocer cuál de ellos es el más barato.

Aunque, en un principio, se utilizará desde un ordenador, cuando se finalice, permitirá conectar con un smartphone.

Los perfiles de usuarios previstos serán:

- Usuario de cualquier edad con conocimientos mínimos de informática.
- Que consideren importante la comparación de precios
- Que valoren la disponibilidad de una lista de compra en su smartphone.
- Que les indique el precio más reducido
- Que les muestre los centros más cercanos
- Que permita llevar un control de las compras realizadas

Aunque en el mercado existen aplicaciones dedicadas a comparar precios, éstas se inician en productos determinados. Esta, sin embargo, se centra en la realización de una lista de compra. Y después, se buscarán los precios y se compararán

25. Marketing y Ventas

Se deben potenciar los aspectos diferenciales de esta aplicación con respecto al resto de existentes en el mercado:

1. Se basa en la creación de una 'lista de compra' en primer lugar y no de un comparativo de precios de inicio.
2. Se debe desarrollar la versión App para Smartphone a continuación para poder disponer de la lista en pantalla y poder realizar la compra física en el establecimiento. Con su visualización por código de barras. Esta versión debe ser diferente de la versión web y enfocada más a la lista que a la selección.
3. Creación y difusión por Facebook principalmente
4. Disponer también de una cuenta de Twitter para relación con usuarios

26. Conclusión.

La idea del proyecto nació de las diferentes consultas realizadas en buscadores de todo tipo.

Siempre me han parecido poco 'usables' en el sentido de practicidad. Sí que, realmente, funcionan en el sentido de comparar el precio de un producto, pero siempre a partir de un elemento determinado. El enfoque de las aplicaciones actuales me ha parecido incorrecto desde el inicio: lo que primero se desea tener es una **lista de compra**. Y, a partir de ahí, comparar los precios. Pero no al contrario.

La realización del proyecto, a nivel conceptual, parece sencilla: manejo de una base de datos con php. Pero, siendo sincero, me ha reportado más problemas de los supuestos en un principio. El Grado en Multimedia que estoy realizando, se ha conseguido con una velocidad de 2 o 3 asignaturas por semestre. Esto conlleva a que, existe mucha diferencia en tiempo desde que se aprenden las materias, hasta que se utilizan en el TFG.

A este punto se añaden otros dos. El primero es que se estudian muchos lenguajes de programación: Java, Javascript, SQL (mySQL, PostGreSQL, PhpMyAdmin), Ruby on Rails, PHP, HTML, HTML5, CSS, Bootstrap, JQuery, C#, *ASP. El segundo, la utilización de un framework (Laravel) que me ha dado muchos más problemas de los planificados. Hasta el punto de decidir no utilizarlo.

Casi todo ha sido nuevo en este proyecto. Incluyendo la contratación de un hosting y subir la aplicación.

De todas formas, ha sido una experiencia muy enriquecedora, ya que ha permitido ver realmente un proyecto acabado y poder aunar todos (casi todos) los conocimientos adquiridos durante el desarrollo de este Grado en Multimedia. In embargo, ha quedado un cierto regusto amargo por no poder implementar todas las ideas surgidas en el desarrollo del proyecto.

Y, por último, aunque la entrega del proyecto se da por terminada para la entrega del TFG, mi intención es intentar desarrollar todas las ideas planteadas en el inicio del proyecto.

Anexo 1. Entregables del proyecto

Lista de archivos entregados y su descripción.

- 1) TABLAS-TFG.XLS – Fichero de definición de los valores de las tablas de la BDD para entorno de pruebas.
- 2) WIREFRAME.RP – Fichero Axure donde se muestran las pantallas de la aplicación
- 3) DIAGRAMA-ER.PDF – Fichero del diagrama Entidad – Relación de la Base de Datos
- 4) NAVEGACION.PDF – Fichero que muestra la navegación del site
- 5) FLOW-CHART.PDF – Fichero que muestra el diagrama de procesos
- 6) DIRECTORIO DE IMÁGENES DE LA BDD – Directorio donde se almacenan las imágenes de los productos y los logos de los supermercados. En formato *zip.
- 7) FICHERO PLANIFICACION – Fichero de Microsoft Project que muestra el diagrama de Gantt del proyecto. En formato *mpp.
- 8) FICHEROS DEL PROYECTO en formato rar
- 9) DOCUMENTACION
 - a. MEMORIA FINAL
 - b. Autoinforme de evaluación
 - c. Presentación en video del proyecto
 - d. Presentacion escrita-visual del proyecto
- 10) BASES DE DATOS – backup en código sql

Para visualizar: [infosami.com /tfg.pkp](http://infosami.com/tfg.pkp)

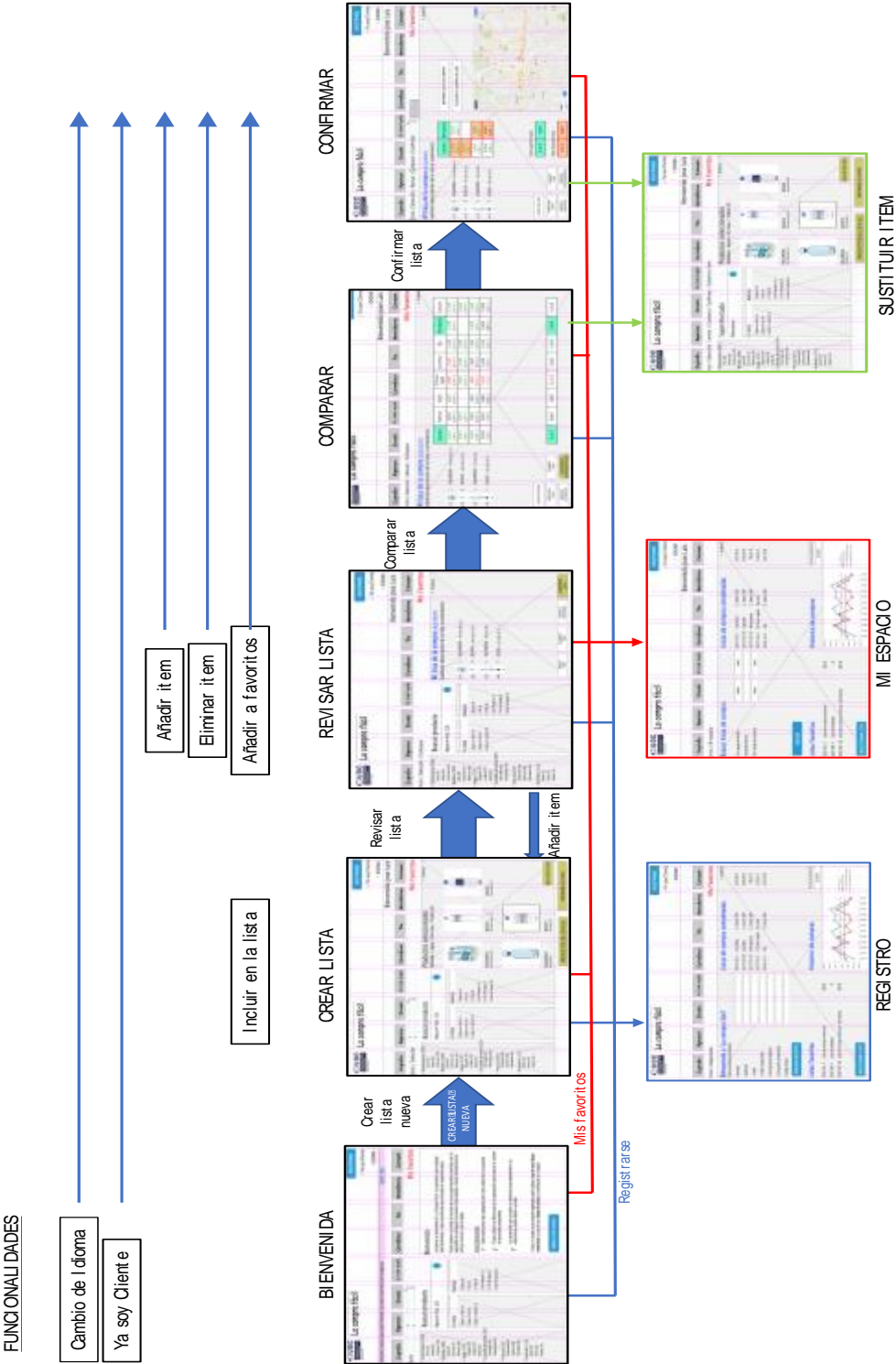
Usuarios creados:

Oli (12345)

Peter (1234)

Antonio (999)

Anexo 2. Diagrama de navegación



Anexo 3. Entidades y Relaciones

Entidades necesarias:

1) Usuario

- ID_Usuario – Entero (8) – Clave única - Autoincremento
- Email – String (50) - Único
- Password – String (16)
- Nombre – String (25)
- Apellidos – String (50)
- CodigoPostal – Entero (6)

2) Sucursal

- ID_Sucursal – Entero (4) – Clave única - Autoincremento
- CodigoSucursal: identifica el código de la sucursal – String (8)
- Dirección de la sucursal – String (50)
- Ciudad – String (25)
- CodigoPostal – Entero (6)
- Latitud – Decimal (3,6)
- Longitud – Decimal (3,6)
- Supermercado_Id – Entero (2) – Identifica a la entidad 'Supermercado' (Relación 1/N) – Clave foránea

3) Supermercado

- ID_Supermercado – Entero (4) – Clave única - Autoincremento
- NombreComercial – String (25)
- UrlAtencion al Cliente – String (100)
- UrlOnLine: acceso a su página on-line – String (100)
- UrlLogotipo: dirección donde se encuentra el logotipo – String (100).

4) Producto

- Id_Producto – Entero (8) – Clave única - Autoincremento
 - Denominación – String (25)
 - Envase: tipo de envase – String (25)
 - Tamaño – String (15)
 - UnidadMedida – String (15)
 - Categoria_Id – Entero (4) – Identifica a la entidad Categoría (Relación N/1) – Clave foránea
- 5) Marca
- Id_Marca – Entero (6) – Clave única - Autoincremento
 - Denominación – String (25)
- 6) Categoría
- Id_Categoria – Entero (4) – Clave única - Autoincremento
 - Denominación – String (25)
- 7) Subcategoría
- Id_Subcategoría – Entero (4) – Clave única - Autoincremento
 - Denominación – String (25)
 - Categoria_Id – Entero (4) – Identifica a la entidad Categoría (Relación N/1) – Clave foránea
- 8) Subdivisión
- Id_Subdivision – Entero (4) – Clave única - Autoincremento
 - Denominación – String (25)
 - Subcategoría_Id – Entero (4) – Identifica a la entidad Subcategoría (Relación N/1) – Clave foránea
- 9) ListaCompra
- Id_Lista – Entero (8) – Clave única - Autoincremento
 - Identificador – String (25)
 - Fecha – Date (10)
 - Usuario_Id – Entero (6) – Identifica a la entidad 'Usuario' – Clave foránea
-

Relaciones entre Entidades

- a) ITEM (Producto / Marca) (N:1)
- b) ESDEUNA (Producto / Categoría) (1: N)
- c) PERTENECE (Categorías / Subcategoría) (1: N)
- d) TIENE (Subcategoría / Subdivisión) (1: N)
- e) SECOMPONE (Supermercado / Sucursal) (1: N)
- f) HACE (Usuario / ListaCompra) (1: N)
- g) ITEMLISTA (Supermercado + Producto) (N: N)
 - Id_ItemLista – Entero (8) – Clave única – Autoincremento
 - Precio – Decimal (5,2)
 - Producto_Id – Entero (8) – Identifica a la entidad 'Producto' – Clave foránea
 - Supermercado_Id – Entero (4) – Identifica a la entidad 'Supermercado' – Clave foránea.
- h) FAVORITO (Usuario / Supermercado) (N: N)
 - IdFavorito – Entero (6) – Clave única - Autoincremento
 - Usuario_Id – Entero (6) – Identifica a la entidad 'Usuario' – Clave foránea
 - Supermercado_Id – Entero (4) – Identifica a la entidad 'Supermercado' – Clave foránea.
- i) SEFORMA (ListaDeCompra / ItemLista) (N: N)
 - IfSeForma – Entero (6) – Clave única - Autoincremento
 - ListaDeCompra_Id – Entero (8) – Identifica a la entidad 'ListaDeCompra' – Clave foránea
 - ItemLista_Id – Entero (8) – Identifica a la entidad 'ItemLista' – Clave foránea.

Anexo 4. Código fuente (extractos)

En este anexo se mostrarán partes relevantes del código y de la aplicación.

(1) Ficheros del Framework Laravel

NOTA REPETIDA – Aunque, al final, no se ha implementado con Laravel, se añaden aquí copias del trabajo realizado

Fichero (.env) de configuración

```

1 APP_ENV=local
2 APP_KEY=base64:fm01Df5+e30HsL0DyYV4Hhsk08jy0RjYKA1r4=
3 APP_DEBUG=true
4 APP_URL=http://localhost
5
6
7 DB_CONNECTION=mysql
8 DB_HOST=127.0.0.1
9 DB_PORT=3306
10 DB_DATABASE=laravel
11 DB_USERNAME=root
12 DB_PASSWORD=root
13
14 PUSHER_HOST=push
15 PUSHER_HOST_PORT=6021
16 PUSHER_APP_KEY=pk_1234567890
17 PUSHER_APP_SECRET=secret
18
19 REDIS_HOST=127.0.0.1
20 REDIS_PASSWORD=null
21 REDIS_PORT=6379
22
23 MAIL_DRIVER=smtp
24 MAIL_HOST=smtp.mailtrap.io
25 MAIL_PORT=2525
26 MAIL_USERNAME=null
27 MAIL_PASSWORD=null
28 MAIL_ENCRYPTION=null
29
30 PUSHER_APP_ID=
31 PUSHER_APP_SECRET=
32 PUSHER_APP_KEY=
    
```

Fichero (web.php) de las rutas de la aplicación (en un estado temprano de desarrollo)

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 Route::get('/', function () {
16     return view('welcome');
17 });
18
19 Route::get('home', function () {
20     return view('welcome',['name' => 'Laravel']);
21 });
22
23 Route::get('login', function () {
24     return view('welcome',['name' => 'Laravel']);
25 });
26
27 Route::get('register', function () {
28     return view('welcome',['name' => 'Laravel']);
29 });
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
    
```

Plantilla de las vistas de la aplicación: se dispone de tres partes en este código.

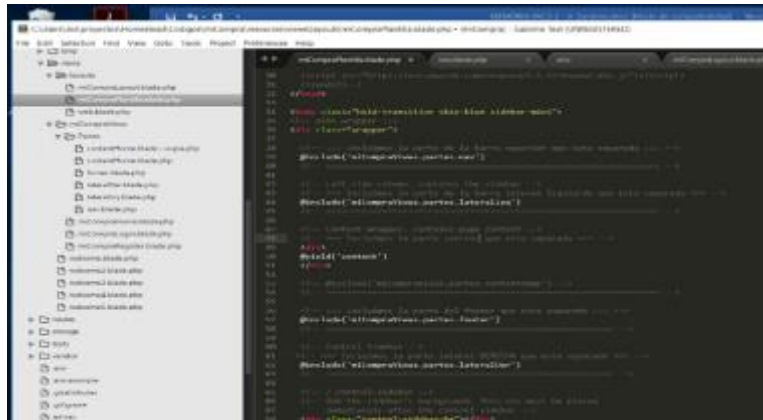
```
<!DOCTYPE html>
<html>
<head>
    1- Links de los css necesarios (bootstrap, Font Awesome, plantilla, fuentes de
        Google fonts, etc.
</head>

<body class="hold-transition skin-blue sidebar-mini">
```

2- Se sitúan las partes en las que se ha dividido la plantilla:

- Barra superior
- Side bar IZQ
- Side bar DER
- Parte central
- Footer

```
<!-- Site wrapper -->
<div class="wrapper">
```



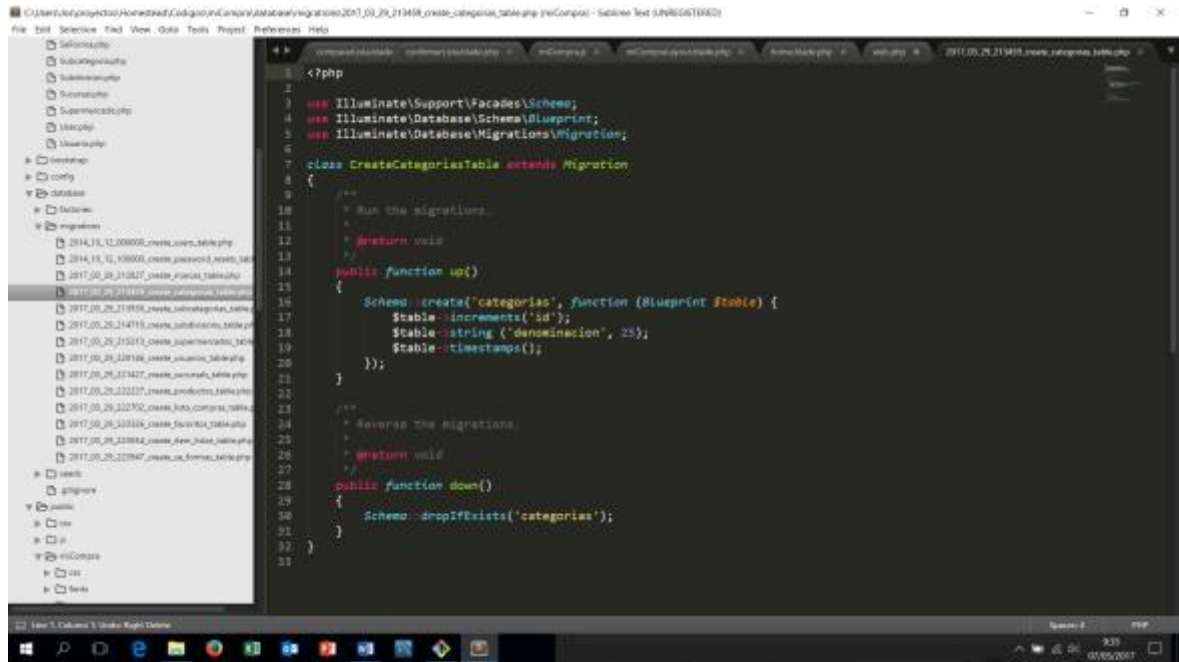
```
<!-- === Incluimos la parte de la barra superior que está separada === -->
    @include('miCompraViews.partes.nav')
<!-- === Incluimos la parte de la barra superior que está separada === -->
    @include('miCompraViews.partes.lateralIzq')
<!-- Content Wrapper. Contains page content -->
<!-- === Incluimos la parte de la barra superior que está separada === -->
    <div>
        @yield('content')
    </div>
<!-- === Incluimos la parte del footer que está separado === -->
    @include('miCompraViews.partes.footer')
<!-- Control Sidebar -->
<!-- === Incluimos la parte lateral DERECHA que está separado === -->
    @include('miCompraViews.partes.lateralDer')
<!-- /.control-sidebar -->
<div class="control-sidebar-bg"></div>
</div>
<!-- ./wrapper -->
```

3- Links de los js necesarios (bootstrap, jquery, plantilla)

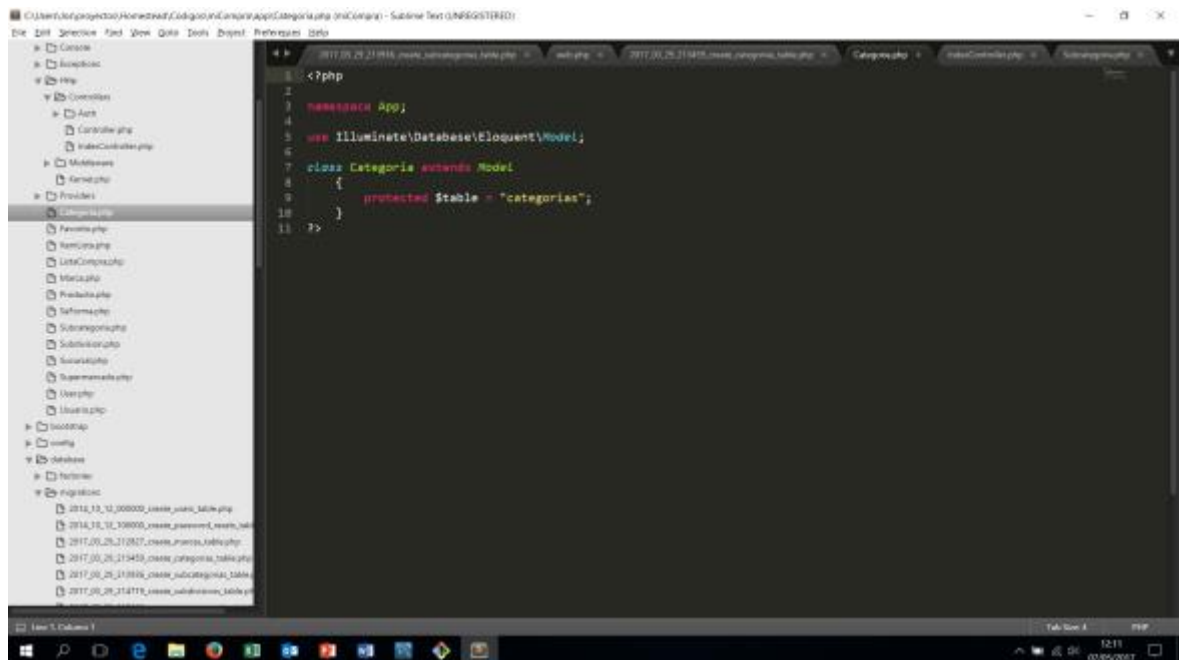
Se puede ver en la imagen anexada la distribución del directorio 'resources/views'

Hacer dinámicos los enlaces de la barra lateral Izquierda, generando los enlaces a partir de la BDD (tablas 'Categorías' y 'Subcategorías').

1 – Las migraciones están creadas



2 – Los modelos están creados



3 – Modificamos el fichero de rutas para que se inicie desde el controlador

```

10 // contains the "web" middleware group. Now create something great!
11
12 //
13
14
15 Route::get('/', 'IndexController@index');
16 Route::get('lista', 'IndexController@lista');
17 Route::get('milista', 'IndexController@milista');
18 Route::get('comparar', 'IndexController@comparar');
19 Route::get('confirmar', 'IndexController@confirmar');
20
21 Route::get('login', function () {
22     return view('miCompraViews.miCompraLogin');
23 });
24 Route::get('register', function () {
25     return view('miCompraViews.miCompraRegister');
26 });
27
28
29
30 /*
31 Route::get('/', function () {
32     return view('miCompraViews.home');
33 });
34 Route::get('home', function () {
35     return view('miCompraViews.home');
36 });
37
38 Route::get('lista', function () {
39     return view('miCompraViews.listaCompra');
40 });
41 Route::get('milista', function () {
42     return view('miCompraViews.milistaCompra');
43 });
44 Route::get('comparar', function () {
45     return view('miCompraViews.compararLista');

```

4 – Generamos ese controlador para que sirva de inicio a la página y creamos los métodos que nos devuelven las vistas.

Definimos unas variables donde cargar los datos de la tabla y las incluimos en la vista:

```

10 use App\Http\Controllers\Controller;
11
12 // Cargamos el modelo de la tabla categorias
13 use App\Categoria;
14 use App\Subcategoria;
15
16 class IndexController extends Controller
17 {
18     public function index() {
19         $categorias = Categoria::all();
20         $subcategorias = Subcategoria::all();
21         return view('miCompraViews.home', ['cats' => $categorias, 'sub' => $subcategorias]);
22     }
23
24     public function lista() {
25         $categorias = Categoria::all();
26         $subcategorias = Subcategoria::all();
27         return view('miCompraViews.listaCompra', ['cats' => $categorias, 'sub' => $subcategorias]);
28     }
29
30     public function milista() {
31         $categorias = Categoria::all();
32         $subcategorias = Subcategoria::all();
33         return view('miCompraViews.milistaCompra', ['cats' => $categorias, 'sub' => $subcategorias]);
34     }
35
36     public function confirmar() {
37         $categorias = Categoria::all();
38         $subcategorias = Subcategoria::all();
39         return view('miCompraViews.confirmarLista', ['cats' => $categorias, 'sub' => $subcategorias]);
40     }
41
42     public function comparar() {
43         $categorias = Categoria::all();
44         $subcategorias = Subcategoria::all();
45         return view('miCompraViews.compararLista', ['cats' => $categorias, 'sub' => $subcategorias]);
46     }
47
48 }

```

(2) Ficheros del de la aplicación:Conexión a la BDD

```

<?php
# Resolvemos la conexion a la base de datos.
$mysqli = new mysqli('localhost', 'root', '', 'infosami_homestead');
if ($mysqli->connect_errno) {
    echo "Error al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
    exit;
}
?>

```

Selección previa

```

# Consulta para las categorias.
$sql1 = "SELECT * FROM categorias";
$result1=$mysqli->query($sql1);
$rowcount1 = mysqli_num_rows($result1);
?>

<!-- ===== Generamos las listas de la BDD ===== -->
<!-- sidebar menu: : style can be found in sidebar.less -->

<ul class="sidebar-menu">
    <?php
        if($result1) {
            while($fila1 = $result1->fetch_object()) { ?>
                <li class="treeview">
                    <a href="#">
                        <i class="fa fa-circle"></i> <span><?php echo
utf8_encode($fila1->denominacion);?></span>

                    </a>
                    <ul class="treeview-menu">
                        <?php
                            # Consulta para las subcategorias.
                            $sql2 = "SELECT * FROM subcategorias";
                            $result2=$mysqli->query($sql2);
                            $rowcount2 = mysqli_num_rows($result2);

                            while($fila2 = $result2->fetch_object()) {
                                if($fila1->id == $fila2->categoria_id) { ?>
                                    <li><a href="listaCompra.php?var1=<?php echo
utf8_encode($fila2->id);?>"><i class="fa fa-circle-o"></i><?php echo
utf8_encode($fila2->denominacion);?>

                                    </a></li>
                                } ?>
                            } ?>
                        </ul>
                    </li>
                </ul>
            } ?>
        } ?>
    </ul>

```


Código HTML de la pantalla de inicio donde se ven los enlaces CSS y los incluye de las partes de la página

```

<!DOCTYPE html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=gb18030">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>miCompra | HOME</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no" name="viewport">
    <!-- Bootstrap 3.3.6 -->
    <link rel="stylesheet" href="/tfg/css/bootstrap.css">
    <!-- Font Awesome -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">
    <!-- DataTables -->
    <link rel="stylesheet" href="/tfg/plugins/datatables/dataTables.bootstrap.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="/tfg/css/AdminLTE.css">
    <!-- Enlace Google Fonts -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Macondo" >
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lobster" >
    <!-- css especificos de la aplicación -->
    <link rel="stylesheet" href="/tfg/css/tfg.css">
    <!-- AdminLTE Skins. Choose a skin from the css/skins
        folder instead of downloading all of them to reduce the load. -->
    <link rel="stylesheet" href="/tfg/css/_all-skins.min.css">
</head>

<body class="hold-transition skin-blue sidebar-mini">
    <!-- Site wrapper -->
    <div class="wrapper">

        <!-- ===== Incluimos la cabecera ===== -->
        <!-- ===== En esta parte se efectua la conexion a la BDD ===== -->
        <?php include 'header.php';?>
        <!-- ===== -->

        <!-- === Incluimos la parte de la barra lateral IZQ que esta separada === -->
        <?php include 'lateralIZQ.php';?>
        <!-- ===== -->

        <!-- ===== Incluimos la parte central del Home ===== -->
        <?php include 'contentHome.php';?>
        <!-- ===== -->

        <!-- ===== Incluimos la parte del footer que esta separada ===== -->
        <?php include 'footer.php';?>
        <!-- ===== -->

        <!-- ===== Incluimos la parte lateral DERECHA que esta separada ===== -->
        <?php include 'lateralDER.php';?>
        <!-- ===== -->

```

Extracto de la parte del Header

```

<?php

    if(!isset($_SESSION)) {
        session_start();
    }

    # Resolvemos la conexion a la base de datos.
    $mysqli = new mysqli('localhost', 'root', '', 'infosami_homestead');
    if ($mysqli->connect_errno) {
        echo "Error al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
        exit;
    }
    $user = isset($_SESSION['user']) ? $_SESSION['user'] : "";
    $url = isset($_SESSION['url']) ? $_SESSION['url'] : "";
    $mail = isset($_SESSION['mail']) ? $_SESSION['mail'] : "";
    $codigo = isset($_SESSION['codigo']) ? $_SESSION['codigo'] : "";

?>

<header class="main-header">

    <!-- Header Navbar: style can be found in header.less -->
    <nav class="navbar navbar-static-top">

        <a href="#" class="sidebar-toggle" data-toggle="offcanvas" role="button">
        <span class="sr-only">Toggle navigation</span>
        </a>

        <div class="navbar-custom-menu">
            <ul class="nav navbar-nav">
                <!-- Messages: style can be found in dropdown.less-->
                <li>
                    <div class="box-body">
                        <a href="/tfgLOGIN/tfgLogin.html" class="btn btn-app">
                            <i class="fa fa-sign-in"></i> Ya soy Cliente
                        </a>
                        <a href="/tfgLOGIN/tfgRegister.html" class="btn btn-app">
                            <i class="fa fa-user-plus"></i> Quiero registrarme
                        </a>
                    </div>
                </li>

                <!-- User Account: style can be found in dropdown.less -->
                <li class="dropdown user user-menu">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                        
                        <span class="hidden-xs"><?php echo $user;?></span>
                    </a>
                </li>
            </ul>
        </div>
    </nav>

```

Extracto de la pantalla de generacion de la lista de Compra. Se puede ver que si no se está registrado, se devuelve el control a la página de login.

```

<!DOCTYPE html>

<?php
session_start();
if(@!$_SESSION['user']) {
    echo '<script language="javascript">window.location="/tfgLOGIN/tfgLogin.html";</script>';
}
?>

<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>miCompra | LISTA</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"
name="viewport">
    <!-- Bootstrap 3.3.6 -->
    <link rel="stylesheet" href="/tfg/css/bootstrap.css">
    <!-- Font Awesome -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">
    <!-- DataTables -->
    <link rel="stylesheet" href="/tfg/plugins/datatables/dataTables.bootstrap.css">
    <!-- Theme style -->

```

Extracto de la pantalla de validación del login

```

<?php
session_start();

# Resolvemos la conexion a la base de datos.
$mysqli = new mysqli('localhost', 'root', '', 'infosami_homestead');
if ($mysqli->connect_errno) {
    echo "Error al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
    exit;
}

$username=$_POST['usuario'];
$password=$_POST['password'];

# Consulta para los usuarios.

if(!$username) {

    echo '<script language="javascript">alert("CONTRASEÑA O PASSWORD INCORRECTOS");</script>';
    echo '<script language="javascript">window.location="../tfg.php";</script>';

}else {
    if(!$password) {

```

```

    }else {
    $sqlL1="SELECT * FROM usuarios WHERE password='$password' AND
nombre='$username'";
    $resultL1=$mysqli->query($sqlL1);
    $rowcountL1 = mysqli_num_rows($resultL1);
    if($fila = $resultL1->fetch_object()){
        header('Location: ../listaCompra.php?var1=20&var2=Bezoya');
        echo $rowcountL1;
        $_SESSION['user']=$username;
        $_SESSION['mail']=$fila->email;
        $_SESSION['codigo']=$fila->codigoPostal;
        $_SESSION['url']=$fila->imagen;
    } else {
        echo '<script language="javascript">alert("CONTRASEÑA
O PASSWORD INCORRECTOS");</script>';
        echo '<script
language="javascript">window.location="../tfg.php";</script>';
    }
}
}

```

Extracto de una parte de la generación del mapa

```

<div class="box-body">
    <div class="row">
        <div class="col-xs-6">
            <a href="#"><button type="button" class="btn btn-block btn-success btn-lg"
onclick="getLocation()">OBTENER POSICION</button></a>
        </div><!-- /.col -->
        <div id="demo" class="col-xs-6">
            <script type="text/javascript">
                var x = document.getElementById("demo");

                function getLocation() {
                    if (navigator.geolocation) {
                        navigator.geolocation.getCurrentPosition(showPosition, showError);
                    } else {
                        x.innerHTML = "Geolocation is not supported by this browser.";
                    }
                }
                // end getLocation

                function showPosition(position) {
                    x.innerHTML = "Latitud: " + position.coords.latitude +
                    "<br>Longitud: " + position.coords.longitude;
                }
                // end showPosition

                function showError(error) {
                    switch(error.code) {
                        case error.PERMISSION_DENIED:
                            x.innerHTML = "User denied the request for Geolocation."
                            break;
                        case error.POSITION_UNAVAILABLE:
                            x.innerHTML = "Location information is unavailable."
                            break;
                        case error.TIMEOUT:
                            x.innerHTML = "The request to get user location timed out."
                            break;
                        case error.UNKNOWN_ERROR:
                            x.innerHTML = "An unknown error occurred."
                            break;
                    }
                }
                // end showError
            </script>
        </div>
    </div>

```

Extracto de una parte de la actualización de la lista de compra con AJAX

Script

```
function addLista() {

    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    } else {
        // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("carrito").innerHTML = this.responseText;
        }
    };
    xmlhttp.open("GET", "crearLista.php", true);

    xmlhttp.send();
}
```

- La función crear Lista debe recoger el array de id's obtenido de la selección e imprimirlo en su div correspondiente.

```
</div><!-- /.box -->
<div id="ocultar" class="box box-danger" style="display: none">
  <div class="box-body ">
    <ul class="nav nav-pills nav-stacked">
      <li style="font-size: 25px"> MI LISTA DE COMPRA</li>
    </ul>
  </div><!-- /.box-body -->

  <div id="carrito" class="box-body no-padding">

  </div><!-- /.box-body -->
```

HTML + PHP

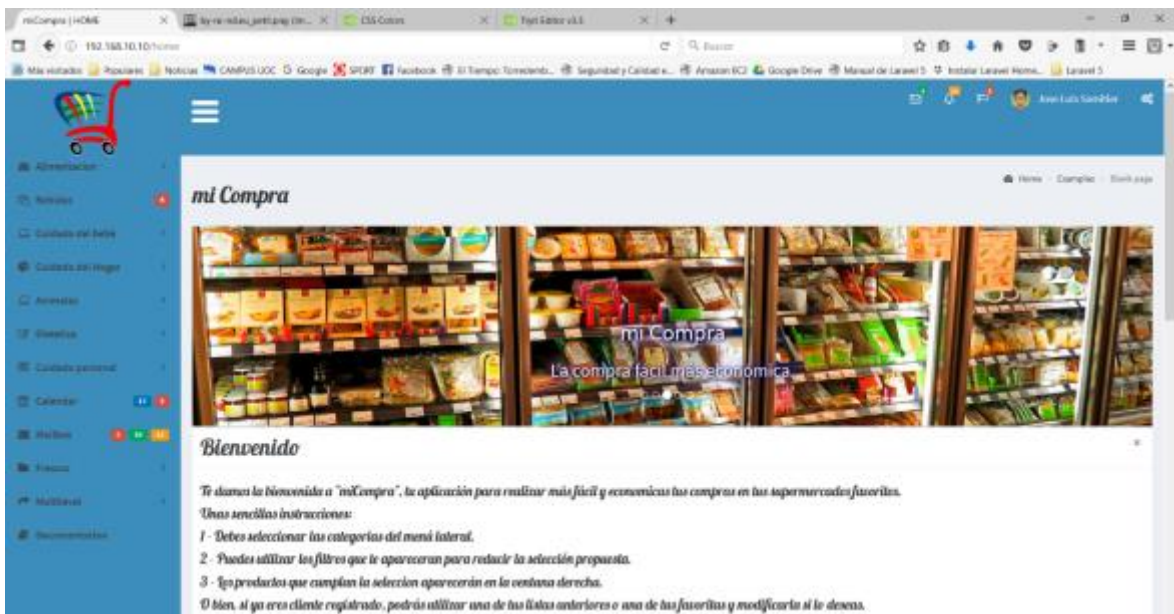
```
<li>
  <a href="#">pid);?>" alt="User Image" style="height: 80px;width: 80px;" ></a>
  <a class="users-list-name" href="#"><?php echo utf8_encode($fila5->filtro);?></a>
  <span class="users-list-name"><?php echo utf8_encode($fila5->envase);?><?php echo utf8_encode($fila5->ptamano);?><?php echo utf8_encode($fila5->pUM);?></span>
  <span class="users-list-name"><?php echo utf8_encode($fila5->pmarca);?></span>
</li>
```

Anexo 5. Capturas de pantalla

Se anexan a continuación las capturas de pantalla del trabajo realizado.

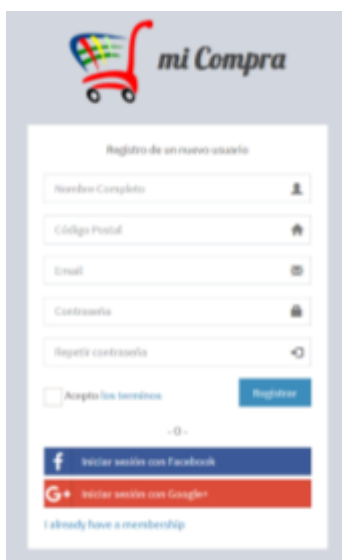
Primero se indican las pantallas de la aplicación:

- 1) Pantalla de INICIO

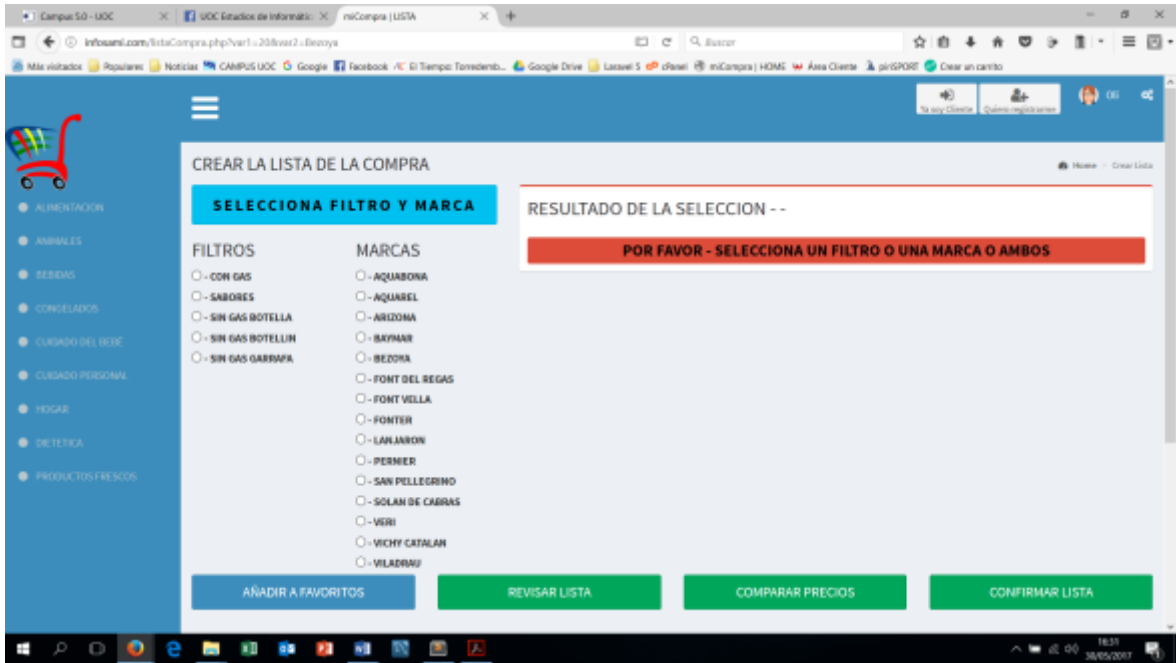


ç

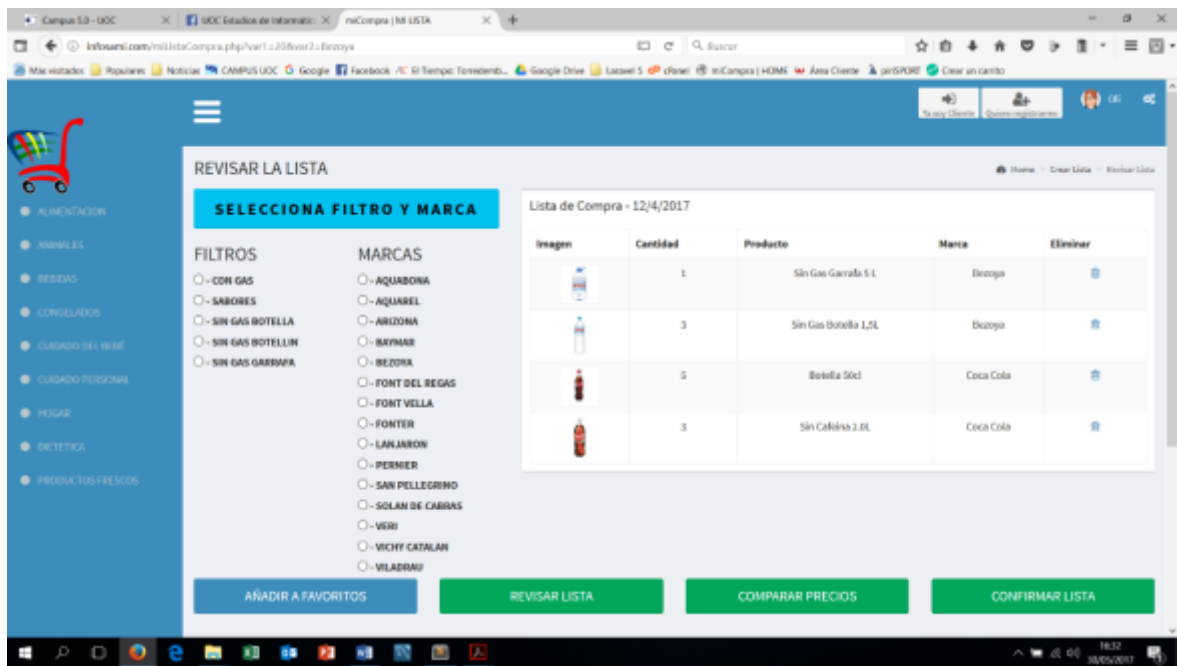
- 2) Pantallas de LOGIN y REGISTRO



3) Pantalla de acceso a crear la LISTA de compra



4) Pantalla de revisión de la lista



5) Pantalla de comparación de precios

COMPARAR PRECIOS

Lista de Compra - 12/4/2017

Imagen	Cant.	Producto	Marca	Caprabo	Hipercor	Eroski	El Corte Inglés	Carrefour	Dia	Mercadona	Consum	ELIMINAR
	1	Sin Gas Garrafa 5 L	Bezoys	2.13€	2.13€	2.13€	0.35€	2.13€	2.13€	0.35€	2.13€	
	3	Sin Gas Botella 1,5L	Bezoys	2.13€	2.13€	2.13€	0.35€	2.13€	2.13€	0.35€	2.13€	
	5	Botella 50cl	Coca Cola	2.13€	2.13€	2.13€	0.35€	2.13€	2.13€	0.35€	2.13€	
	3	Sin Cabello 2,0L	Coca Cola	2.13€	2.13€	2.13€	0.35€	2.13€	2.13€	0.35€	2.13€	

AÑADIR A FAVORITOS REVISAR LISTA COMPARAR PRECIOS CONFIRMAR LISTA

6) Pantalla de confirmación de la lista

CONFIRMAR LA LISTA

Lista de Compra - 12/4/2017

Imagen	Cant.	Producto	Marca	Caprabo	Condis
	1	Sin Gas Garrafa 5 L	Bezoys	2.13€	2.13€
	3	Sin Gas Botella 1,5L	Bezoys	2.13€	2.13€
	5	Botella 50cl	Coca Cola	2.13€	2.13€
	3	Sin Cabello 2,0L	Coca Cola	2.13€	2.13€

IMPRIMIR LISTA COMPRAR ON-LINE Servicio Postal

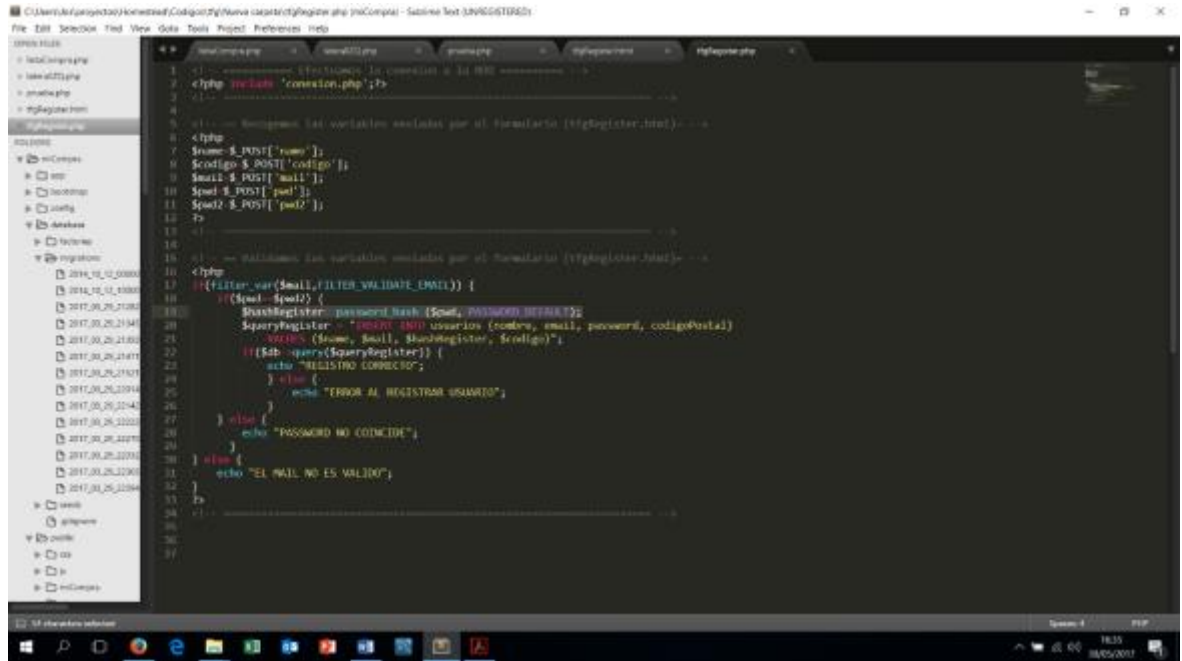
OBTENER POSICION

Direccion: _____ Ciudad: _____

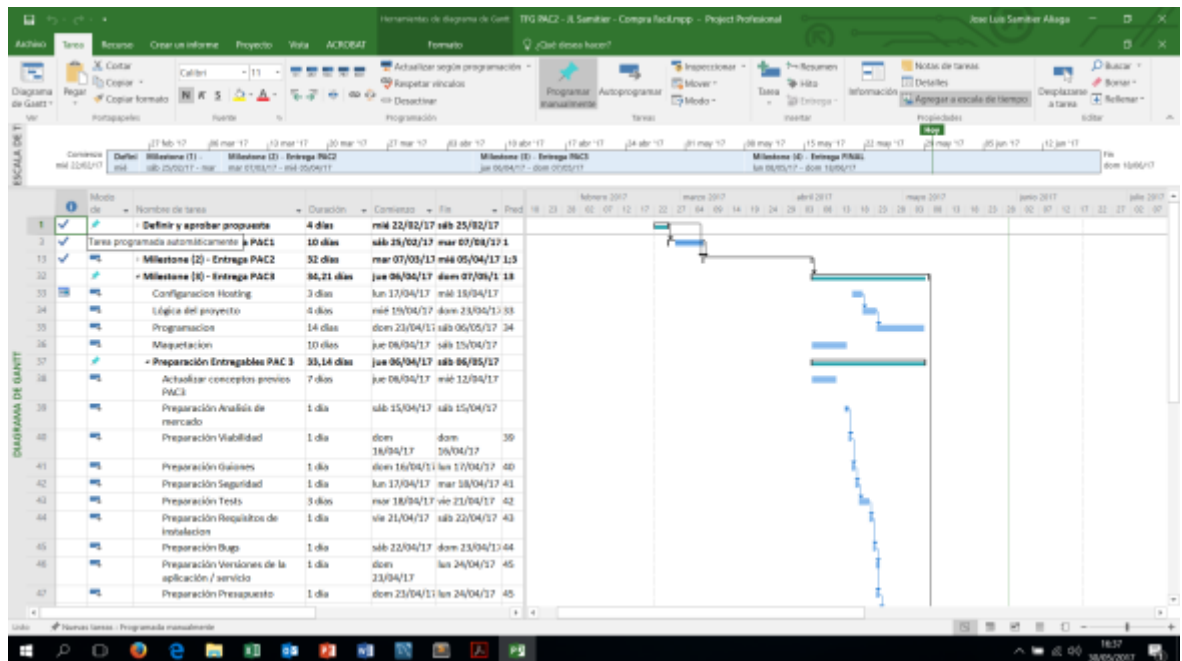
Provincia: _____ País: _____ LOCALIZAR

AÑADIR A FAVORITOS REVISAR LISTA COMPARAR PRECIOS CONFIRMAR LISTA

7) Editor de textos 'Sublimetext'



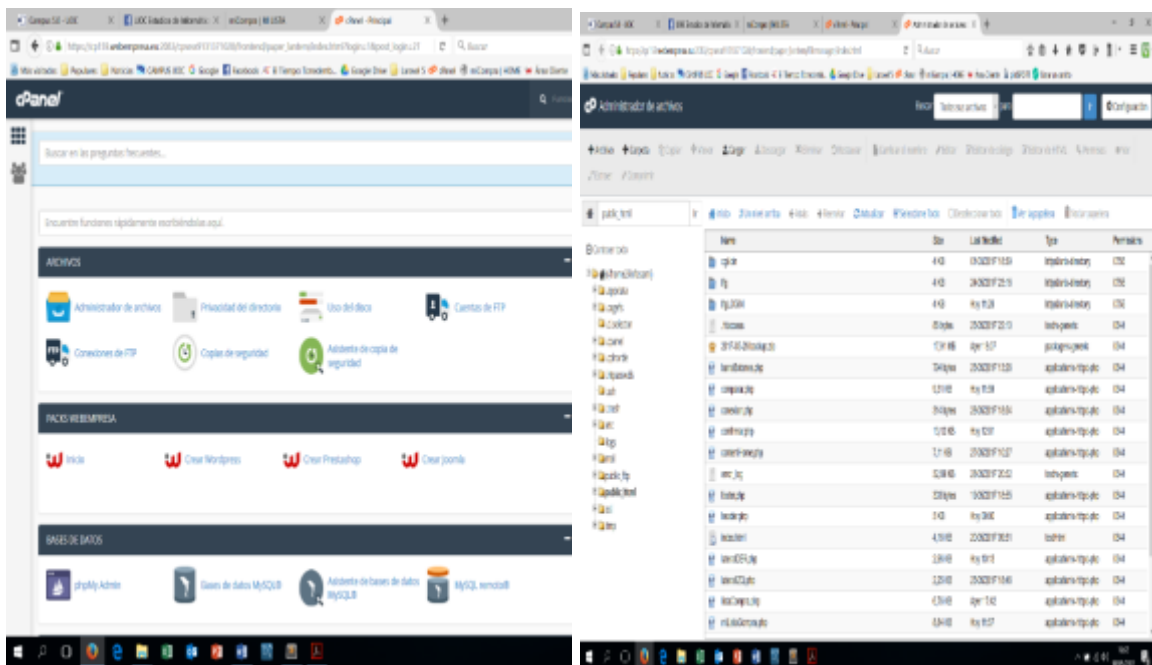
8) Microsoft Project – Diagrama de Gantt del proyecto



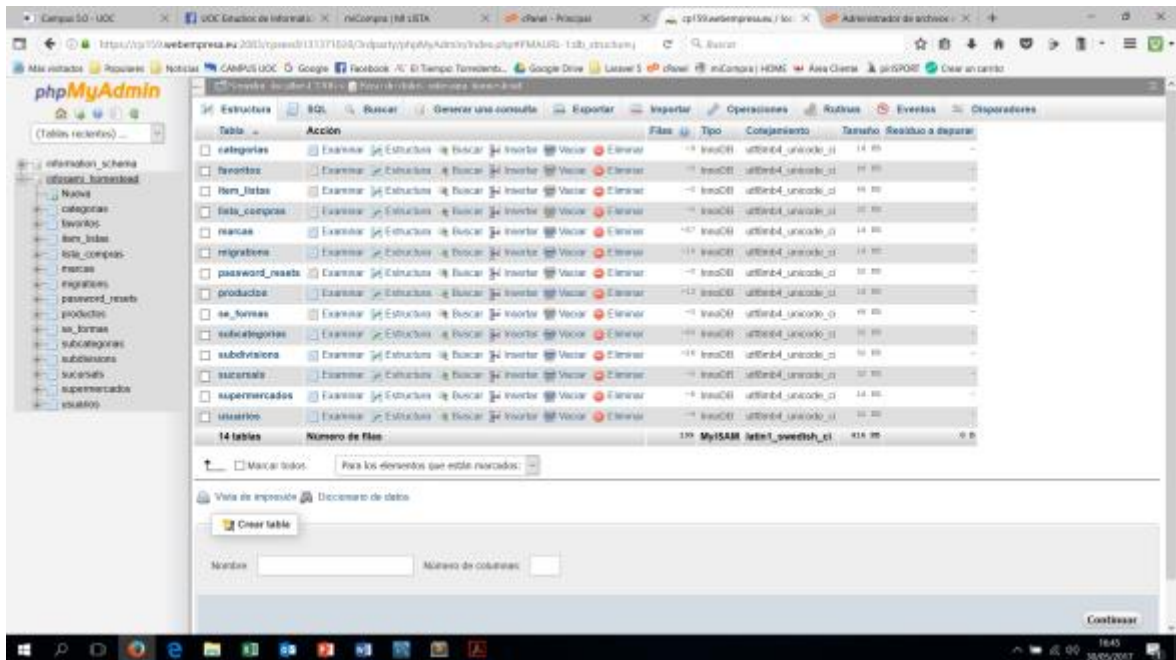
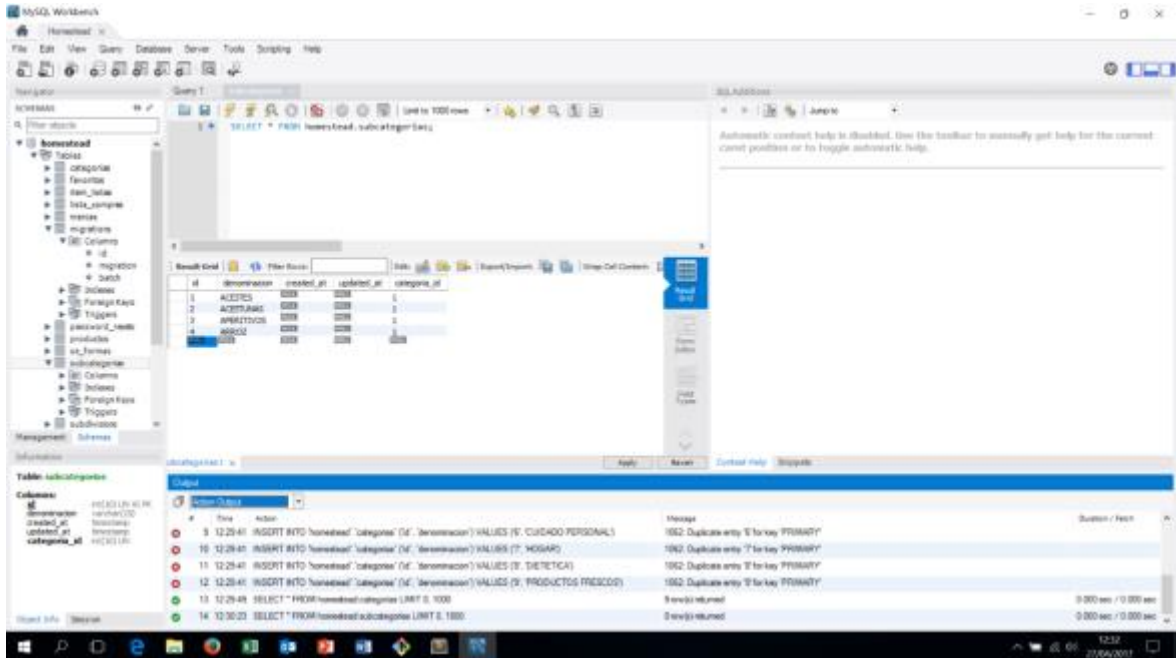
9) Pantalla de inicio del servidor local Wampserver para desarrollo



10) Pantalla del control del hosting y de administración (Webempresa)



11) Pantallas de administración de la BDD



Anexo 6. Resumen ejecutivo

Nombre comercial:

miCompra

Resumen comercial:

Básicamente es una aplicación que gestiona la creación de una lista de compra y que indica los precios más competitivos en los supermercados de referencia. .

Modelo de negocio:

Este producto viene a solventar una falta de usabilidad de los productos actuales.

Mercado:

Este producto se dirige a todo tipo de público, pero los usuarios de 20 a 40 años son los que, a priori, obtendrán más beneficio.

Competencia:

En el mercado existen diferentes alternativas, pero todas enfocadas a la comparación de precio como punto de partida. El aspecto diferencial de esta propuesta reside en que su punto de inicio es la creación de una lista de compra, que después se valida.

Plan de márketing:

Se basará en un buen posicionamiento SEO y una difusión por las redes sociales.

Inversión inicial y costes a corto plazo:

La inversión inicial se centra en el diseño y realización de la página. Se estiman unos 5.000€ de inversión.

Proyección económica:

Este proyecto no plantea clientes que paguen una cuota.

Su rentabilidad pasará por la inclusión de cierta publicidad relacionada con el tema.

DAFO:

- Debilidades:
Su punto crítico será la usabilidad. Especialmente en smartphones..
La generación y el mantenimiento de la tabla de productos
- Fortalezas:
Es un producto innovador.
- Amenazas:
La confianza de los usuarios en los datos suministrados.
- Oportunidades:
El público principal al que va dirigido utiliza mucho las redes sociales. Si se consigue esta confianza, el avance puede ser muy rápido.

Anexo 7. Bibliografía

MATERIALES DIDACTICOS UOC

- (1) Introducción a la Programación en PHP (PID 00151997) – Jaume Gil
- (2) Programación Web. Javascript (PID 00149620) – Vicent Moncho Mas
- (3) Diseño de Interfaces Multimedia. Material docente de la UOC (PID_00159830) – Tona Monjo Palau
- (4) Diseño de Bases de Datos (PID 00150710) – Piero Berni Millet y Dídac Gil de la Iglesia
- (5) Introducción al diseño de bases de datos (PID P06/M2109/02150) – Dolors Costal Costa
- (6) Laboratorio de PHP y MySQL (PID 00155713) – Piero Berni Millet y Dídac Gil de la Iglesia
- (7) Gestión de Proyectos (PID 00153527) – José Ramón Rodríguez
- (8) Ingeniería del Software (PID 00230156) – Jordi Pradel Miquel y Jose Raya Martos
- (9) Fundamentos de JQuery (PID 00235949) – Rebecca Murphey
- (10) Introducción a la programación web avanzada (PID 00172704) – Jordi Sánchez Cano
- (11) Administración de sistemas GNU Linux (PID 00212806) – Josep Jorba Esteve y Remo Suppi Boldrito.
- (12) Linux avanzado (PID 00212464) - Josep Jorba Esteve y Remo Suppi Boldrito.
- (13) Seguridad y Calidad en Servidores Web – Daniel Torrico Robledo, Héctor Alonso Martín y Marco Marín Martínez
- (14) Diseño y programación orientada a objetos (PID 00235262) – David García Solórzano
- (15) Seguridad en BDD (PID 00191660) - José María Alonso Cebrián, Vicente Díaz Sáez, Antonio Guzmán Sacristán, Pedro Laguna Durán, y Alejandro Martín Bailón
- (16) Vulneraciones de seguridad (PID 00178964) - José María Alonso Cebrián, Sergio Castillo Pérez, Joaquín García Alfaro, Antonio Guzmán Sacristán, Jordi Herrera Joancomartí, Pedro Laguna Durán, Alejandro Martín Bailón, Guillermo Navarro Arribas y Sergi Robles Martínez

CURSOS REALIZADOS (Barcelona Activa - Cibernarium)

- (1) Introducción a lenguaje Javascript.
- (2) Máquinas Virtuales: instalación de diferentes sistemas virtuales
- (3) Introducción a PHP
- (4) Introducción a BDD relacionales con MySql

- (5) Uso de PHP en el desarrollo de recursos digitales
- (6) Introducción a CSS3
- (7) Crear un proyecto responsive con Bootstrap
- (8) T07 Introducción a la programación de aplicaciones con HTML5 (curso Microsoft)
- (9) Introducción a JQuery para diseñadores
- (10) Introducción a la programación con Windows.NET
- (11) Construye y personaliza tu web con Wordpress
- (12) Infografía: aprende a representar la información gráficamente
- (13) Herramientas para la visualización de datos
- (14) Introducción a la seguridad en entornos wi-fi corporativos
- (15) Técnicas de ciberseguridad en sistemas y redes informáticas
- (16) Administra un sistema Linux (1): instalación, configuración y acceso a los recursos
- (17) Administra un sistema Linux (2): gestión de servicios, procesos y aspectos avanzados.
- (18) Crea y administra BDD con MySQL con phpMyAdmin
- (19) Introducción a la programación orientada a objetos con Java

TUTORIALES

1) Bootstrap:

- https://librosweb.es/libro/bootstrap_3/
- <https://www.w3schools.com/bootstrap/>
- Curso completo de Bootstrap 3 – videos de YouTube:
https://www.youtube.com/watch?v=nug1pMke-y4&list=PLhSj3UTs2_yWTKvu1Aq3xUhzIJNBZ3MFW

2) Laravel:

- Curso de Laravel 5.0 – Raúl Palacios YouTube
https://www.youtube.com/watch?v=togljDT95wo&list=PLlddmSRJEJ0u-5Nv2k6W8Vhe0wUP_7H5W
- Curso de Laravel 5.2 + Bootstrap YouTube
https://www.youtube.com/watch?v=Lvsk9bEdS80&list=PLu40Vfz0GBbrj_QVMBA28TJHP54kDKeVR
- (<https://laravel.com/docs/5.4/homestead>)

- Tutorial (<https://www.youtube.com/watch?v=k9xQBKzpOjk>)
- Tutorial:
<https://desarrolloweb.com/articulos/videotutorial-instalar-homestead-laravel-windows.html>

REALIZACION DE LA MEMORIA

- (1) Normas para la redacción del proyecto fin de carrera

https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0ahUKEwjEh9m52fPSAhXHcRQKHxD0C6QQFgghMAA&url=http%3A%2F%2Fwww4.tecnun.es%2Fasignaturas%2Fpfc%2FPlantillas%2FNormas_Redaccion_PFC.doc&usq=AFQjCNE1ib4zgURCcVmkwmHZgW6HocGfHQ

- (2) Normas de estilo de proyectos fin de carrera

<https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwjEh9m52fPSAhXHcRQKHxD0C6QQFggnMAE&url=https%3A%2F%2Fwww.uclm.es%2Farea%2Fproyectosingenieria%2Fimages%2Fpresen.doc&usq=AFQjCNF4et0QBQNI0giULgGk5ZfcX-hnYQ>